Fourier Series: Review

A function or signal $x(t)$ is called *periodic* with period T if $x(t+T)=x(t)$. All “typical” *periodic* function $x(t)$ with period T can be developed as follows

Equation:

$$x\left(t\right)=\frac{a_0}{2}+\sum_{k=1}^{\infty}\ \ a_k\cos\left(k\frac{2\pi}{T}t\right)+b_k\sin\left(k\frac{2\pi}{T}t\right)$$

where the coefficients are computed as follows:

Equation:

$$a_k=\frac{2}{T}\int_{-T/2}^{T/2}x\left(t\right)\cos\left(k\frac{2\pi}{T}t\right)dt\quad\left(k=0,1,2,\dots\right)$$

Equation:

$$b_k=\frac{2}{T}\int_{-T/2}^{T/2}x\left(t\right)\sin\left(k\frac{2\pi}{T}t\right)dt\quad\left(k=1,2,3,\dots\right)$$

The natural interpretation of [link] is as a decomposition of the signal $x(t)$ into individual oscillations where a_k indicates the amplitude of the even oscillation $\cos\left(k\frac{2\pi}{T}t\right)$ of frequency k/T (meaning its period is T/k), and b_k indicates the amplitude of the odd oscillation $\sin\left(k\frac{2\pi}{T}t\right)$ of frequency k/T. For an audio signal $x(t)$, frequency corresponds to how high a sound is and amplitude to how loud it is. The oscillations appearing in the Fourier decomposition are often also called harmonics (first, second, third harmonic etc).

Note: one can also integrate over $[0,T]$ or any other interval of length T. Note also, that the average value of $x(t)$ over one period is equal to $a_0/2$.

Complex representation of Fourier series

Often it is more practical to work with complex numbers in the area of Fourier analysis. Using the famous formula

Equation:

$$e^{j\alpha}=\cos\left(\alpha\right)+j\sin\left(\alpha\right)$$

it is possible to simplify several formulas at the price of working with complex numbers. Towards this end we write

Equation:

$$a\cos\left(\alpha\right)+b\sin\left(\alpha\right)=a\frac{1}{2}\left(e^{j\alpha}+e^{-j\alpha}\right)+b\frac{1}{2j}\left(e^{j\alpha}-e^{-j\alpha}\right)=\frac{1}{2}(a-bj)e^{j\alpha}+\frac{1}{2}(a+bj)e^{-j\alpha}$$

From this we observe that we may replace the cos and sin harmonics by a pair of exponential harmonics with opposite frequencies and with complex amplitudes which are conjugate complex to each other. In fact, we arrive at the more simple *complex Fourier series*:

Equation:

$$x\left(t\right)=\sum_{k=-\infty}^{\infty}X_k e^{j2\pi kt/T}\qquad\text{with}\quad X_k=\frac{1}{T}\int_0^T x\left(t\right)e^{-j2\pi kt/T}dt$$

Note that X is complex, but x is real-valued (the imaginary parts of all the terms in $\sum X_k e^{j2\pi kt/T}$ add up to zero; in other words, they cancel each other out). The absolute value of X_k gives the amplitude of the complex harmonic with frequency k/T (meaning its period is T/k); the argument of X_k provides the phase difference between the complex harmonics. If x is even, X_k is real for all k and all harmonics are in phase.

To verify [link] note that by [link] and [link] we have for positive k

Equation:

$$X_k=\frac{1}{T}\int_0^T x\left(t\right)[\cos(2\pi kt/T)-j\sin(2\pi kt/T)]dt=\frac{1}{2}(a_k-b_kj).$$

For negative k we note that $X_{-k}=X_k^*$ by [link], where $()^*$ denotes the conjugate complex. By [link], the X_k are exactly as they are supposed to be.

Properties

- Linearity: The Fourier coefficients of the signal $z(t)=cx(t)+y(t)$ are simply

Equation:

$$Z_k=cX_k+Y_k$$

- Change of frequency: The signal $z(t)=x(\lambda t)$ has the period T/λ and has the same Fourier coefficients as $x(t)$ — but they correspond to different frequencies f:

Equation:

$$Z_k\ _{f=\frac{k}{T/\lambda}}=X_k\ _{f=\frac{k}{T}}\qquad\text{since}\qquad z\left(t\right)=x\left(\lambda t\right)=\sum_{k=-\infty}^{\infty}X_k e^{j2\pi\lambda kt/T}=\sum_{k=-\infty}^{\infty}Z_k e^{j2\pi t\frac{k}{T/\lambda}}$$

The equation on the right allows to read off the Fourier coefficients and to establish $Z_k=X_k$. (For an alternative computation see Comment 1)

Comment 1

Equation:

$$\begin{aligned} Z_k &= \frac{1}{T/\lambda}\int_0^{T/\lambda} z\left(t\right)e^{-2\pi jkt/(T/\lambda)}dt = \frac{\lambda}{T}\int_0^{T/\lambda} x\left(t\lambda\right)e^{-2\pi jkt\lambda/T}dt \\ &= \frac{1}{T}\int_0^{T} x\left(s\right)e^{-2\pi jks/T}ds = X_k \end{aligned}$$

- Shift: The Fourier coefficients of $z(t) = x(t+d)$ are simply

Equation:

$$Z_k = X_k e^{j2\pi kd/T}$$

The modulation is much more simple in complex writing then it would be with real coefficients. For the special shift by half a period, i.e., $d = T/2$ we have $Z_k = X_k e^{j\pi k} = \left(-1\right)^k X_k$.

- Derivative: The Fourier series of the derivative of $x(t)$ with development [link] can be obtained simply by taking the derivative of [link] term by term:

Equation:

$$x'\left(t\right) = \sum_{k=-\infty}^{\infty} X_k \cdot k\frac{2\pi j}{T} \cdot e^{j2\pi kt/T}$$

Short: when taking the derivative of a signal, the complex Fourier coefficients get multiplied by $k\frac{2\pi j}{T}$. Consequently, the coefficients of the derivative decay slower.

Examples

- The pure oscillation (containing only one real but two complex frequencies)
  Equation:

  $$x\left(t\right) = \sin\left(2\pi t/T\right) \qquad X_1 = -X_{-1} = \frac{j}{2}$$

  or $B_1 = 1/2 = -B_{-1}$, or $b_1 = 1$, and all other coefficients are zero. This formula can be obtained without computing integrals by noting that $\sin\left(\alpha\right) = \left(e^{j\alpha} - e^{-j\alpha}\right)/\left(2j\right) = \left(j/2\right)\left(e^{-j\alpha} - e^{j\alpha}\right)$ and setting $\alpha = 2\pi t/T$.
- Functions which are *time-limited*, i.e., defined on a finite interval can be periodically extended. Example with $T = 2\pi$:
  Equation:

$$x\left(t\right)=\begin{array}{ll} 1 & \text{for }\ 0<t<\pi \\ -1 & \text{for }\ -\pi<t<0 \\ c & \text{for }\ t=0,\pi \end{array} \qquad b_k=2B_k=\frac{4}{\pi k}\quad \text{for odd }\ k\geq 1$$

and all other coefficients zero. Note that c is any constant; the value of c does not affect the coefficients b_k. We have for $0<t<\pi$

Equation:

$$1=\frac{4}{\pi}\left(\sin\left(t\right)+\frac{1}{3}\sin\left(3t\right)+\frac{1}{5}\sin\left(5t\right)+\ldots\right)$$

Note that for $t=0$ the value of the series on the right is 0, which is equal to $x(0-)+x(0+)$, the middle of the jump of $x(t)$ at 0, no matter what c is. Similar for $t=\pi$.

Discrete Fourier Transform

Discrete Fourier Transform

The Discrete Fourier Transform, from now on DFT, of a finite length sequence $(x_0, \dots, x_{K-1})$ is defined as

Equation:

$$\text{mover}_k = \overset{K-1}{\underset{n=0}{}} x_n e^{-2\pi jk\frac{n}{K}} \qquad (k = 0, \dots, K-1)$$

To motivate this transform think of x_n as equally spaced samples of a T-periodic signal $x(t)$ over a period, e.g., $x_n = x\left(nT/K\right)$. Then, using the Riemann Sum as an approximation of an integral, i.e.,

Equation:

$$\overset{K-1}{\underset{n=0}{}} f \quad \frac{nT}{K} \quad \frac{T}{K} \simeq \overset{T}{\underset{0}{}} f\left(t\right)dt$$

we find

Equation:

$$x_k = \overset{K-1}{\underset{n=0}{}} x \quad \frac{nT}{K} \quad e^{-2\pi j\frac{nT}{K}k/T} \simeq \frac{K}{T} \overset{T}{\underset{0}{}} x\left(t\right)e^{-2\pi jtk/T}dt = KX_k$$

Note that the approximation is better, the larger the sample size K is.

Remark on why the factor K in [link]: recall that X_k is an average while x_k is a sum. Take for instance $k = 0$: X_0 is the average of the signal while x_0 is the sum of the samples.

From the above we may hope that a development similar to the Fourier series [link] should also exist in the discrete case. To this end, we note first that the DFT is a linear transform and can be represented by a matrix multiplication (the “exponent” T means transpose):

Equation:

$$x_0, ..., x_{K-1}{}^T = DFT_K \cdot (x_0, ..., x_{K-1})^T.$$

The matrix DFT_{K} possesses K lines and K rows; the entry in line k row n is $e^{-2\pi jkn/K}$. A few examples are

Equation:

$$DFT_1 = (1) \quad DFT_2 = \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix} \quad DFT_4 = \begin{matrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{matrix}$$

The rows are orthogonal[footnote] to each other. Also, all rows have length[footnote]$\sqrt{K}$. Finally, the matrices are symmetric (exchanging lines for rows does not change the matrix). So, the multiplying DFT with its conjugate complex matrix $(DFT_K)^*$ we get K times the unit matrix (diagonal matrix with all diagonal elements equal to K).

The scalar product for complex vectors $x = (x_1, x_2, ..., x_K)$ and $y = (y_1, y_2, ..., y_K)$ is computed as

Equation:

$$x \cdot y = x_1 y_1^* + x_2 y_2^* + ... + x_K y_K^*,$$

where a^* is the conjugate complex of a. Orthogonal means $x \cdot y = 0$.

Length is computed as

$$||x|| = \sqrt{x \cdot x} = \overline{x_1 x_1^* + x_2 x_2^* + ... + x_K x_K^*} = \overline{x_1{}^2 + |x_2{}^2 + ... + |x_K|^2}$$

.

Inverse DFT

From all this we conclude that the inverse matrix of DFT_K is $IDFT_K = (1/K) \cdot (DFT_K)^*$. Since $(e^{-\alpha})^* = e^{\alpha}$ we find

Equation:

$$x_n = \frac{1}{K} {}_{k=0}^{K-1} x_k e^{2\pi jk\frac{n}{K}} \qquad (n = 0, ..., K-1)$$

Spectral interpretation, symmetries, periodicity

Combining [link] and [link] we may now interpret x_k as the coefficient of the complex harmonic with frequency k/T in a decomposition of the discrete signal x_n; its absolute value provides the amplitude of the harmonic and its argument the phase difference.

If x is even, x_k is real for all k and all harmonics are in phase.

Using the periodicity of $e^{2\pi jt}$ we obtain $x_n = x_{n+K}$ when evaluating [link] for arbitrary n. Short, we can consider x_n as equally-spaced samples of the T-periodic signal $x(t)$ over any interval of length T:

Equation:

$$x_k = \mathop{}\limits_{n=-K/2}^{K/2-1} x_n e^{-2\pi jk\frac{n}{K}}.$$

Similarly, x_k is periodic: $x_k = x_{k+K}$. Thus, it makes sense to evaluate x_k for any k. For instance, we can rewrite [link] as

Equation:

$$x_n = \frac{1}{K} \mathop{}\limits_{n=-K/2}^{K/2-1} x_k e^{2\pi jk\frac{n}{K}}$$

Since x_k corresponds to the frequency k/T, the period K of x_k corresponds to a period of K/T in actual frequency. This is exactly the sampling frequency (or sampling rate) of the original signal (K samples per T time units). Compare to the spectral repetitions.

However, the period T of the original signal x is nowhere present in the formulas of the DFT (cpre. [link] and [link]). Thus, if nothing is known about T, it is assumed that the sampling rate is 1 (1 sample per time unit), meaning that $K = T$.

FFT

The Fast Fourier Transform (FFT) is a clever algorithm which implements the DFT in only $K \log(K)$ operations. Note that the matrix multiplication would require K^2 operations.

Matlab implements the FFT with the command `fft(x)` where x is the input vector. Note that in Matlab the indices start always with 1! This means that the first entry of the Matlab vector x, i.e. $x(1)$ is the sample point $x_0 = x(0)$. Similar, the last entry of the Matlab vector x is, i.e. $x(K)$ is the sample point $x_{K-1} = x((K-1)T/K) = x(T - T/K)$.

Fourier Integral

Fourier Integral

Continuous-time signals which are not periodic can still be understood as superpositions of pure oscillations where now all frequencies are present in the signal. The coefficients of the oscillations can be computed as follows:
Equation:

The representation as a superposition takes then the following form:
Equation:

We call the *Fourier transform* of and write also instead of to indicate clearly which signal has been transformed. The “Fourier spectrum”, or simply the *spectrum*, or also the “power spectrum” of the signal is the squared amplitude . This is the function usually plotted, while the phase of is not shown. Nevertheless, the plots are usually —and erroneously— labeled with instead of (see [link]).

A signal is called *bandlimited* if its Fourier transform is zero for high frequencies, i.e. for large . Similarly we say that a signal is *time-limited* if it is zero for large times, i.e., for large . By Heisenberg's principle a bandlimited signal can not be time-limited. Since bandlimited signals are of great importance, there is a need to study signals which are not time-limited and, thus, the Fourier integral.

Properties

- Linearity:
  Equation:

- Convolution
  Equation:

- Change of time scale
  Equation:

  — — — —

- Translation in time and frequency
  Equation:

- Symmetries and Fourier pairs The symmetry of [link] and [link] leads one to consider and as a Fourier pair. Indeed, the Fourier transform of is almost : . Clearly, the symmetry is not perfect since is in general complex, while is real. However: If is symmetric, i.e. then is real-valued, and vice versa!

In summary: *Symmetric real signals have symmetric real Fourier transforms and vice versa*. As we will see below, they also possess the same energy.

Energy and Power

Energy and Power

The *energy* of a continuous-time signal $x(t)$ is given as
Equation:

$$||x||^2 := \int_{-\infty}^{\infty} x^2\,(t)dt$$

Plancherel's theorem says (for more information see Comment 2): If the signal $x(t)$ has finite energy then its Fourier transform $X(f)$ has the *same* energy:
Equation:

$$||X\,(f)||^2 = \int_{-\infty}^{\infty} |X|^2\,(f)df = \int_{-\infty}^{\infty} x^2\,(t)dt = ||x||^2 \qquad [\text{finite energy case}]$$

Comment 2 Plancherel theorem is a result in harmonic analysis, first proved by Michel Plancherel. In its simplest form it states that if a function f is in both $L_1\,(\mathbb{R})$ and $L_2\,(\mathbb{R})$, then its Fourier transform is in $L_2\,(\mathbb{R})$; moreover the Fourier transform map is isometric. This implies that the Fourier transform map restricted to $L_1\,(\mathbb{R}) \cap L_2\,(\mathbb{R})$ has a unique extension to a linear isometric map $L_2\,(\mathbb{R}) \to L_2\,(\mathbb{R})$. This isometry is actually a unitary map.

Periodic signals have of course infinite energy; therefore, one introduces the *power* P_x of the signal $x(t)$, which is the average energy over one period. Energy is measured in Joule, power is measured in Watt=Joule/Sec.

The analog of Plancherel's theorem is Parseval's theorem which applies to T-periodic signals and says
Equation:

$$P = P_x = \frac{1}{T}\int_{-T/2}^{T/2} |x\,(t)|^2 dt = \sum_{k=-\infty}^{\infty} |X_k|^2 \qquad [\text{periodic case}]$$

We may derive Parseval's theorem as follows, using [link] and $|a|^2 = a \cdot a^*$:
Equation:

$$
\begin{aligned}
P_x &= \frac{1}{T}\int_{-T/2}^{T/2} |x(t)|^2 dt = \frac{1}{T}\int_{-T/2}^{T/2} \sum_{k=-\infty}^{\infty} X_k e^{j2\pi kt/T} {}^{2} dt \\
&= \frac{1}{T}\int_{-T/2}^{T/2} \sum_{k=-\infty}^{\infty} X_k e^{j2\pi kt/T} \quad \sum_{n=-\infty}^{\infty} X_n e^{j2\pi nt/T} {}^{*} dt \\
&= \frac{1}{T}\int_{-T/2}^{T/2} \sum_{k=-\infty}^{\infty} X_k e^{j2\pi kt/T} \quad \sum_{n=-\infty}^{\infty} X_n^{*} e^{-j2\pi nt/T} dt \\
&= \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} X_k X_n^{*} \frac{1}{T}\int_{-T/2}^{T/2} e^{j2\pi(k-n)t/T} dt \\
&= \sum_{k=-\infty}^{\infty} X_k X_k^{*} = \sum_{k=-\infty}^{\infty} |X_k|^2
\end{aligned}
$$

Here, we used that ${}_{-T/2}^{T/2} e^{j2\pi(k-n)t/T}$ equals T when $k = n$ (since $e^0 = 1$), but equals 0 when $k \neq n$ since (since $e^{js} = \cos(s) + j\sin(s)$, which are integrated over several periods).

A similar computation can be carried out for Plancherel's equation. However, some difficulties arise due to the integrals over infinite intervals (see Comment 3 below). Also, a justification of Plancherel could be given by performing a limit of infinite period in Parseval's equation (see Comment 4 below).

For finite discrete signals the analog is simply the fact, that DFT is unitary up to a stretching factor. More precisely, the matrix DFT_K leaves angles intact and stretches length by $\sqrt{K}$. Intuitively, one may think of the DFT as a rotation and a stretching. In other words, to perform a DFT simply means to change the coordinate system into a new one, and to change length measurement by a factor $\sqrt{K}$. Thus:

Equation:

$$
P_x = \frac{1}{K}\sum_{n=1}^{K} x_n^2 = \frac{1}{K^2}\sum_{k=1}^{K} x_k {}^{2} = \frac{1}{K}P_x \qquad [\text{discrete case}]
$$

Note that the DFT Fourier coefficients are complex numbers; thus, the absolute value has to be taken (for a complex number a we have $|a|^2 = a \cdot a^*$, which is usually different from a^2—unless a is by chance real valued).

Comment 3 A “hand-waving” argument for Plancherel's theorem runs as follows, using [link] and $|a|^2 = a \cdot a^*$:

Equation:

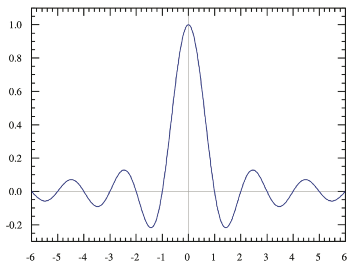
$$\begin{aligned}
||x(t)||^2 &= \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \,^{2}\, dt \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad \int_{-\infty}^{\infty} X(g) e^{j2\pi gt} dg \,^{*}\, dt \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad \int_{-\infty}^{\infty} X(g)^* e^{-j2\pi gt} dg \;\; dt \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(f) X(g)^* \int_{-\infty}^{\infty} e^{j2\pi t(f-g)} dt \; df dg \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(f) X(g)^* \delta(f-g) df dg \\
&= \int_{-\infty}^{\infty} X(f) X(f)^* dt = \int_{-\infty}^{\infty} |X(f)|^2 dt = ||X||^2
\end{aligned}$$

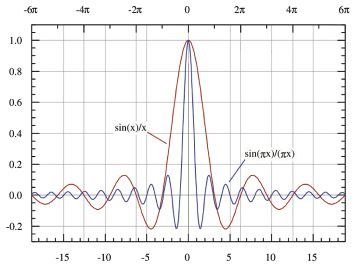
Thereby, the step $^{\infty}_{-\infty} e^{j2\pi t(f-g)} dt = \delta(f-g)$ would require some more care, but we content ourselves here with this intuitive computation.

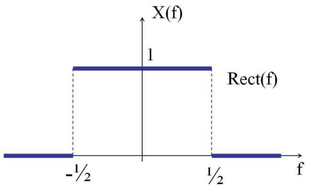
An important example of a band-limited signal: the sinc-function.

Sinc and the un-normalized version in comparison.

The power spectrum of sinc is Rect(f).







Comment 4 With Parseval's equation established one may provide a derivation of Plancherel's formula which is more convincing, tho more difficult to make rigorous. Assume that the signal $x(t)$ is time-limited, say defined on $-T/2 < t < T/2$. Its Fourier-series coefficients are $X_k = 1/T \quad {}^{T/2}_{-T/2} x\,(t)e^{-j2\pi kt/T}dt$. Comparing to the Fourier transform we find thus $T \cdot X_k = X\,(k/T)$. We may interpret x as being periodically extended and use Parseval's equation which says $P_x = \quad |X_k|^2$; this allows the following computation

Equation:

$$\begin{aligned} ||x||^2 &= \int_{-T/2}^{T/2} |x\,(t)|^2 dt = T \cdot P_x = T \sum_{k=-\infty}^{\infty} \quad X_k \quad {}^2 = \sum_{k=-\infty}^{\infty} \frac{1}{T} T^2 |X_k|^2 \\ &= \sum_{k=-\infty}^{\infty} \frac{1}{T} |X\,(k/T)|^2 \simeq \int_{-\infty}^{\infty} |X\,(t)|^2 dt \end{aligned}$$

In the last step we use a Riemann sum approximation of the integral. It remains to extend this to signals which are not time-limited. The approximations made are more believable than the step using the Dirac delta (see Comment 3 above). Although more intuitive and believable, these arguments are harder to make rigorous than making the computation of Comment 3 rigorous by passing through a correct, direct computation of the integrals avoiding the Dirac delta.

Examples

- The pure oscillation (containing only one frequency)
  Equation:

  $$\sin\left(2\pi t/T\right) \qquad\qquad \mathscr{F}\{\sin\left(2\pi t/T\right)\}\left(f\right) = \frac{j}{2}\left(\delta\left(-1\right) - \delta\left(1\right)\right)$$

  This formula can be obtained without computing integrals by noting that $\sin\left(x\right) = \left(e^{jx} - e^{-jx}\right)/\left(2j\right) = \left(j/2\right)\left(e^{-jx} - e^{jx}\right)$. Its power is $P = 1/2$ (see [link]).
- The perfect *low (frequency) pass* function:
  Equation:

  $$\operatorname{sinc}\left(t\right) := \frac{\sin\left(\pi t\right)}{\pi t} \qquad \mathscr{F}\{\operatorname{sinc}(t)\}\left(f\right) = \operatorname{Rect}\left(f\right) = \begin{cases} 1 & \text{if } \; -1/2 < f < 1/2 \\ 0 & \text{else.} \end{cases}$$

  and more general (to pass exactly the frequencies $f \in]-f_c, f_c[$)
  Equation:

  $$2f_c \cdot \operatorname{sinc}\left(2f_c t\right) \qquad \mathscr{F}\{2f_c\operatorname{sinc}\left(2f_c t\right)\}\left(f\right) = \operatorname{Rect}\left(\frac{f}{2f_c}\right) = \begin{cases} 1 & \text{if } \; -f_c < f < f_c \\ 0 & \text{else.} \end{cases}$$

  Both, x and X are symmetric and real. Plancherel's formula allows to compute the energy of the sinc:
  Equation:

  $$||\operatorname{sinc}\left(t\right)||^2 = \int_{-\infty}^{\infty} X^2\left(f\right)df = \int_{-1/2}^{1/2} 1df = 1.$$

  More generally, the energy of $2f_c \cdot \operatorname{sinc}\left(2f_c t\right)$ amounts to $2f_c$. Note that the sinc is not time-limited; it can't be by the Heisenberg principle since it is bandlimited.
- The Dirac function $\delta(t)$ is often symbolically written as
  Equation:

  $$\delta\left(t\right) = \begin{cases} \;\infty'' & \text{if } \; t = 0 \\ 0 & \text{else.} \end{cases} \qquad \mathscr{F}\{\delta(t)\}\left(f\right) \equiv 1$$

  Clearly, the Dirac function is not really a function, and it has infinite energy. However, most manipulations work fine also for $\delta(t)$. This is again an illustration of the Heisenberg principle. The Dirac function is the extreme case which is sharply located in time, but has no characteristic frequency (all frequencies are present with equal strength). The properties of the Dirac function are best understood in terms of integrals:
  Equation:

  $$\int_u^v g\left(t\right)\delta\left(t\right)dt = \begin{cases} g(0) & \text{if } \; u < 0 < v \\ 0 & \text{else} \end{cases}.$$

  As a special case, the convolution with a function *g* is again *g*:
  Equation:

  $$\{\delta * g\}\left(a\right) = \int_{-\infty}^{\infty} \delta\left(a - t\right)g\left(t\right)dt = \int_{-\infty}^{\infty} \delta\left(t\right)g\left(a - t\right)dt = g\left(a\right)$$

  short: $\delta * g(t) = g(t)$. For the shifted Dirac function $\delta_b\left(t\right) = \delta\left(t - b\right)$ we have
  Equation:

$$\{\delta_b * g\}(a) = \int_{-\infty}^{\infty} \delta_b(t) g(a-t)dt = \int_{-\infty}^{\infty} \delta(t-b) g(a-t)dt = \int_{-\infty}^{\infty} \delta(s) g(a-(s+b))ds = g(a-b)$$

short, convolution with δ_b produces a shift by b: $\delta_b * g(t) = g(t-b)$.

- Double exponential:
  Equation:

  $$x(t) = e^{-|t|} \qquad X(f) = \frac{2}{1+4\pi^2 f^2}$$

  Note that x and its Fourier transform X are real and symmetric. The power spectrum is $|X(f)|^2 = 4/\left(1+4\pi^2 f^2\right)^2$. Since x is not differentiable at 0, the Fourier transform X decays somewhat slowly: high frequencies are quite strong in this signal in order to make the sharp peak at 0. With this example, we may compute the energy directly:
  Equation:

  $$||e^{-|t|}||^2 = \int_{-\infty}^{\infty} \left(e^{-|t|}\right)^2 dt = 2\int_0^{\infty} e^{-2t} dt = -e^{-2t}\Big|_0^{\infty} = 1.$$

- One-sided Exponential:
  Equation:

  $$x(t) = \begin{cases} e^{-t} & \text{if } t > 0 \\ 0 & \text{else.} \end{cases} \qquad X(f) = \frac{1}{1+2\pi j f}$$

  The Fourier transform X is complex with power spectrum $|X(f)|^2 = 1/\left(1+4\pi^2 f^2\right)$. Since x is not even continuous at 0, the Fourier transform X decays even slower than for the double exponential: high frequencies are even stronger in this signal in order to make the jump at 0. With this example, we may verify Plancherel's theorem:
  Equation:

  $$||x||^2 = \int_0^{\infty} \left(e^{-t}\right)^2 dt = \int_0^{\infty} e^{-2t} dt = -e^{-2t}/2\Big|_0^{\infty} = 1/2.$$

  Equation:

  $$||\frac{1}{1+2\pi j f}||^2 = \int_{-\infty}^{\infty} \frac{1}{1+4\pi^2 f^2} df = \frac{1}{2\pi} \arctan(2\pi t)\Big|_{-\infty}^{\infty} = \frac{1}{2\pi}(\pi/2 - (-\pi/2)) = 1/2.$$

- The Gaussian Kernel is practically invariant under the Fourier transform (see Comment 5)
  Equation:

  $$x(t) = e^{-t^2/2} \qquad X(f) = \sqrt{2\pi} e^{-(2\pi f)^2/2}$$

  Here, it is easy to verify $||x|| = ||X||$ via a substitution $t = 2\pi f$. The computation is somewhat harder and yields $||x||^2 = \sqrt{\pi} \simeq 1.7725$.

Comment 5 From Probability theory, we know that (see “characteristic function of a Gaussian distribution”)
Equation:

$$\int e^{itf} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt = e^{-f^2/2}$$

Now replace f by $2\pi f$ and multiply with $\sqrt{2\pi}$ to find the Fourier transform. For the energy:

Equation:

$$||x||^2 = \int_{-\infty}^{\infty} \left(e^{-t^2/2}\right)^2 dt = \sqrt{\pi}\frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty} e^{-t^2} dt = \sqrt{\pi}$$

where we use in the last step, that $\frac{1}{\sqrt{\pi}}e^{-t^2}$ constitutes the probability density of a Gaussian variable with variance $1/2$, and thus integrates to 1.

- The Mexican Hat (also called Ricker Wavelet in Geophysics) is the negative second derivative of the Gaussian:
  Equation:

  $$x(t) = \left(1 - t^2\right)e^{-t^2/2} \qquad X(f) = 4\pi^2\sqrt{2\pi}f^2 e^{-(2\pi f)^2/2}$$

  Both, the Gaussian kernel and the Mexican hat are very useful since they are well located both in space and in frequency (see [link]), meaning that the main portion of their energy stems from a narrow range of locations as well as a narrow range of frequencies. Thus, the may be used as low-pass, respectively band-pass filters. The Mexican hat is a wavelet; wavelets are used to determine which frequencies contribute the main portion of the energy at a specific time. To this end, a wavelet needs to be well localized in time as well as in frequency.
- The Dirac Comp (peigne de Dirac) of step τ
  Equation:

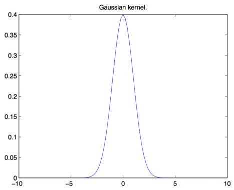
  $$\tau \cdot \mathit{\Delta}_\tau(t) := \tau \sum_n \delta(t - n\tau) \qquad \mathscr{F}\{\tau \cdot \mathit{\Delta}_\tau\}(f) = \sum_{k=-\infty}^{\infty} \delta(f - k/\tau)$$

  To verify this formula, we choose the integration interval $[-\tau/2, \tau/2]$ and write
  Equation:

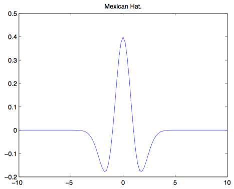
  $$X_k = \frac{1}{\tau}\int_{-\tau/2}^{\tau/2} \tau \cdot \mathit{\Delta}_\tau(t) e^{-j2\pi kt/\tau} dt = \sum_n \int_{-\tau/2}^{\tau/2} \delta(t - n\tau) e^{-j2\pi kt/\tau} dt = \int_{-\tau/2}^{\tau/2} \delta(t) e^{-j2\pi kt/\tau} dt = 1$$

  The Fourier-representation becomes
  Equation:

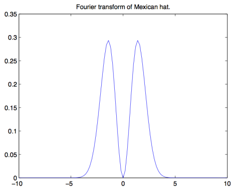
  $$\tau \cdot \mathit{\Delta}_\tau(t) = \sum_{k=-\infty}^{\infty} e^{j2\pi kt/\tau}$$



The Gaussian kernel. Its Fourier transform has the identical shape.



The Mexican hat is the negative second derivative of the Gaussian kernel.



Fourier transform of the Mexican hat.

This formula will be crucial in the reconstruction formula in the Nyquist-Shannon Theorem. Note: One should not try to evaluate $\tau \cdot \varDelta_{\tau}\left(t\right)$; indeed, it is not really a function, since it is formed by Dirac terms. Even its power is infinite.

Estimation of Spectrum and Power via DFT

Spectrum for finite energy signals

The Fourier integral is intended for non-periodic signals, since the periodicity will lead to problems with the convergence of the integral. In fact, the Fourier integral [link] is defined rigorously only for signals with finite energy (see next section below). Recall that finite energy signals can never be periodic. For an alternative way of defining $X(f)$ for periodic signals see the next bullet just below).

However, in order to make the DFT also useful for non-periodic finite energy signals, we can consider any signal $x(t)$ as being restricted to the interval $[-L/2, L/2]$ and then extended periodically. Denote this new L-periodic signal by $\widetilde{x}\,(t)$ with Fourier coefficients $\widetilde{X}_k$. When L is large, comparing [link] with [link] gives

Equation:

$$\widetilde{X}_k = \frac{1}{L}\int_{-L/2}^{L/2} \widetilde{x}\,(t) e^{-j2\pi kt/L} dt \simeq \frac{1}{L}\int_{-\infty}^{\infty} x\,(t) e^{-j2\pi kt/L} dt = \frac{1}{L} X \quad \frac{k}{L} \quad .$$

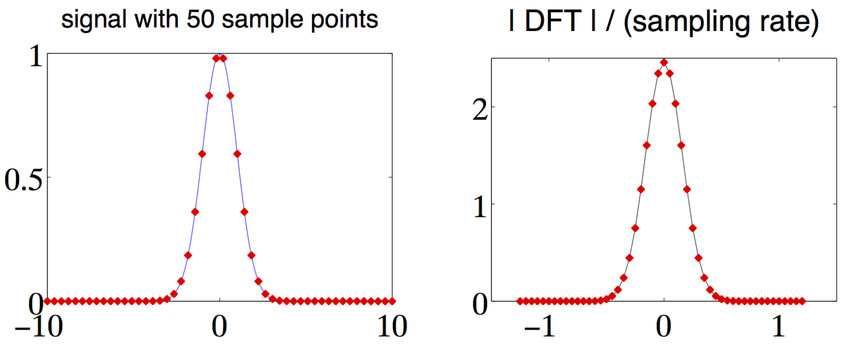
The notation $\widetilde{X}_k$ instead of X_k should remind the reader, that x itself is not periodic.

If not the entire signal x but only some K samples $x_n = x\,(nL/K)$ of the signal are given, we may use the DFT as in [link] and combine it with [link] to get for $k = -K/2, . . , K/2 - 1$

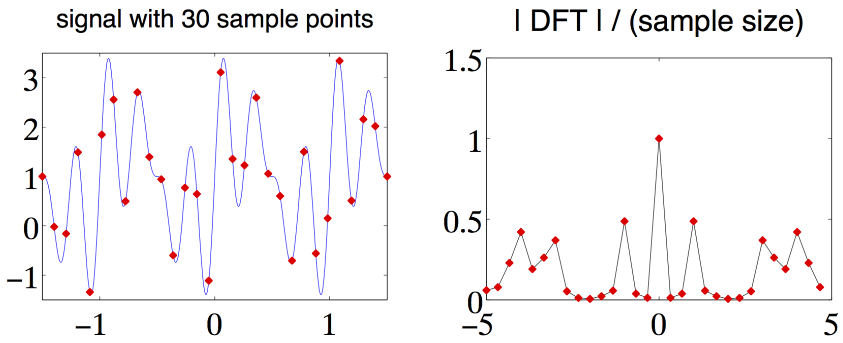
Equation:

$$\text{finite energy:} \qquad X \quad \frac{k}{L} \quad \simeq L \cdot \widetilde{X}_k \simeq \frac{L}{K} \widehat{x}_k \ .$$

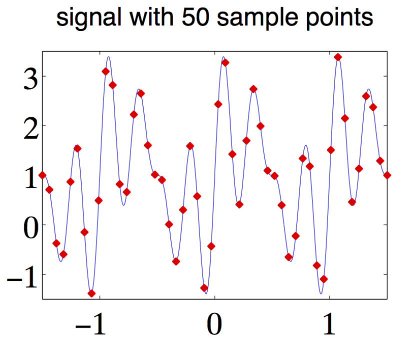
Note that this approximation improves in quality as L becomes larger; the factor L/K indicates that K would have to be made proportionally larger at the same time. For an illustration see [link], [link], and [link], left column, with $L = 20$, $K = 50$. Imagine that we reduce L to half, i.e. to $L^* = 10$, leading to a new signal x^*. This would mean to discard half of the samples, thus leading to $K^* = 25$. Since the discarded samples are practically zero, using [link] shows that $\widehat{x^*}_k \simeq \widehat{x}_{2k}$. This agrees with [link] since $L^*/K^* = L/K$ and $X\left(k/L^*\right) = X\,(2k/L)$.

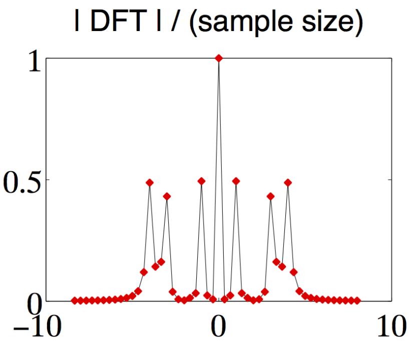


Spectral Estimation via DFT. A finite energy signal (Gaussian kernel).



Spectral Estimation via DFT. The sampled periodic signal $x(t) = 1 + \sin(2\pi t) + \sin(3 \cdot 2\pi t) + \sin(4 \cdot 2\pi t)$ (period $T = 1$, length $L = 3$). The larger the sample-size K, the better are the locations (0, 1, 3 and 4) and amplitudes (1, 0.5, 0.5 and 0.5) of the delta peaks of X approximated by $|\widehat{x}_k|/K$.





Spectral Estimation via DFT. The sampled periodic signal $x(t) = 1+ \sin\left(2\pi t\right) + \sin\left(3 \cdot 2\pi t\right) + \sin\left(4 \cdot 2\pi t\right)$ (period $T = 1$, length $L = 3$). The larger the sample-size K, the better are the locations (0, 1, 3 and 4) and amplitudes (1, 0.5, 0.5 and 0.5) of the delta peaks of X approximated by $\left|\hat{x}_k\right|/K$.

Spectrum for periodic signals

For T-periodic signals, the Fourier integral [link] does not converge in the usual sense. One can still define a Fourier transform $X(f)$ consisting of Dirac delta functions:

Equation:

$$X\left(f\right) = \sum_k X_k \delta\left(f - k/T\right) \simeq \sum_k \frac{1}{K}\hat{x}_k \delta\left(f - k/T\right).$$

With this setting we have $x\left(t\right) = \int X\left(f\right)e^{j2\pi ft}df$ also for periodic signals. Note that the Delta-functions in [link] are "infinitely large" at $f = k/T$, in agreement with [link] where L should ideally be "infinitely large".

However, "infinitely large" values are not useful when one wants to read off an amplitude. Comparing [link] and [link] we see that the DFT $\hat{x}_k$ should be normalized differently for non-periodic finite energy signals than for periodic signals. Dividing the DFT by the *sampling rate* for *non-periodic* signals provides an estimation of the Fourier transform via [link]. Dividing the DFT by the *sample size* for *periodic* signals provides an estimation of the peak heights of the Dirac delta functions in [link] (see [link] and [link]).

Energy and Power from Sampling

First of all, note that the energy of a periodic signal is infinite because there are infinitely many periods with the same energy. On the other hand, the power of a finite energy signal is zero since it is the average energy of an infinitely long interval.

Short, only one measure is meaningful for a signal: either the power or the energy, depending on context. Nevertheless, we may compare power and energy of the various ways of representing a signal: as a finite energy signal, as a signal on a given finite interval (extending it periodically) or through its samples.

Comparing the power of a T-periodic signal with its samples over an interval of length L, where L is an integer multiple of the period T, we find in the frequency domain (using [link] and [link])
Equation:

$$P_x\left(\text{cont}\right)=\sum_{k=-\infty}^{\infty}\left|X_k\right|^2\simeq\sum_{k=-K/2}^{K/2}\left|X_k\right|^2\simeq\sum_{k=-K/2}^{K/2}\frac{1}{K^2}\left|\widehat{x}_k\right|^2=P_x\left(\text{sampled}\right)$$

Computing in the time domain (using a Riemann Sum) we arrive at the same conclusion:
Equation:

$$P_x\left(\text{cont}\right)=\frac{1}{L}\int_{-L/2}^{L/2}\left|x\left(t\right)\right|^2dt\simeq\frac{1}{L}\sum_{n=-K/2}^{K/2}\frac{L}{K}\left|x\quad n\frac{L}{K}\quad\right|^2=\frac{1}{K}\sum_{n=-K/2}^{K/2}\left|x_n\right|^2=P_x\left(\text{sampled}\right)$$

Comparing the energy for finite energy signals and for their samples on an interval of length L we find
Equation:

$$\text{Energy}=\left|\left|x\right|\right|^2\simeq\int_{-L/2}^{L/2}\left|x\left(t\right)\right|^2dt=L\cdot P_x\left(\text{cont}\right)\simeq L\cdot P_x\left(\text{sampled}\right)$$

where we used [link] in the last step.

In summary, using sufficiently many samples taken over sufficiently large intervals should allow to compute the power or energy of signals in either of the three representations, thereby approximating power with energy per time over a long interval.

As an important conclusion we note that the power of a periodic signal should not depend on the number of samples taken, at least approximatively, and as long as sufficient many samples are taken. We will later make precise how many samples are sufficient. Similar for the energy of a finite energy signal.

As a final remark we note that the name “power spectrum” used for $\left|X\left(f\right)\right|^2$, $\left|X_k\right|^2$ and $\left|\widehat{x}_k\right|^2$ is appropriate since they indicate how the power is distributed of the spectrum of frequencies, i.e., which frequencies contribute much and which less to the power.

Sampling: Review

Let us consider a signal $x(t)$ which is uniformly sampled meaning that the samples are $x_n = x\left(n\tau\right)$ $(n = ... - 2, -1, 0, 1, 2, ...)$. The sampling frequency $f_e = 1/\tau$ is the number of samples per time unit.

In practical applications, the signal will be sampled only over an interval, say of length L and into K samples. Then $\tau = L/K$ as for discrete signals.

Spectral copies (also called spectral repetitions or images)

The sampled signal can be written as the Dirac comb of step τ modulated by $x(t)$

Equation:

$$x_e\left(t\right) = x\left(t\right) \cdot \tau \mathit{\Delta}_\tau\left(t\right) \quad = \quad \sum_{n=-\infty}^{\infty} \tau x\left(n\tau\right) \cdot \delta\left(t - n\tau\right)$$

The factor τ is added to obtain a Fourier transform that does not depend on τ and to preserve an averaging property (the integral of x_e provides a good approximation of the integral of x since $\int x_e\left(t\right)dt = \sum_{n=-\infty}^{\infty} \tau x\left(n\tau\right)$ is the Riemann sum approximating the $\int x\left(t\right)dt$ as a sum of rectangles of width τ.)

Using [link] we may write

Equation:

$$x_e\left(t\right) = x\left(t\right) \cdot \tau \mathit{\Delta}_\tau\left(t\right) \quad = \quad \sum_{k=-\infty}^{\infty} x\left(t\right) \cdot e^{j2\pi kt/\tau}$$

Note, that [link] is *not* a Fourier representation of $x_e\left(t\right)$ (the “coefficients” $x(t)$ of the sinusoids depend on t instead of k). Computing the Fourier transform of the sampled signal $x_e\left(t\right)$ once more, using [link], we find

Equation:

$$X_e(f) = \sum_{k=-\infty}^{\infty} \left\{x(t)e^{j2\pi kt/\tau}\right\}(f) = \sum_{k=-\infty}^{\infty} X(f-k/\tau)$$

We observe that the Fourier transform, and also the spectrum of the sampled signal is periodic with period $f_e = 1/\tau$, the sample rate (see [link] right). However, we see now that X_e is composed of copies of X, shifted by a multiple of the period, i.e., shifted by $k/\tau = k \cdot f_e$. These are called *spectral copies*.

Sampling and the discrete Fourier transform

Finite energy signals: Using [link] again we find the Fourier transform of the sampled signal $x_e(t)$

Equation:

$$X_e(f) = \sum_{n=-\infty}^{\infty} \tau x(n\tau) \quad \{\delta(t-n\tau)\}(f) = \tau \sum_{n=-\infty}^{\infty} x_n \cdot e^{-j2\pi fn\tau}$$

Setting now $f = k/L$ and recalling $\tau = L/K$ we get approximately

Equation:

$$\text{finite energy:} \qquad X_e(k/L) \simeq \tau \sum_{n=-K/2}^{K/2} x_n \cdot e^{-j2\pi(k/L)n(L/K)} = \tau\widehat{x}_k.$$

This agrees with [link] for finite energy signals and is actually valid for all k, since both sides are periodic with period K. However, it is only useful if x is of finite energy, similar to [link].

For periodic signals we could write X and X_e as sums of Dirac delta functions as in [link]. Doing so, [link] confirms that also periodic signals possess spectral copies when sampled.

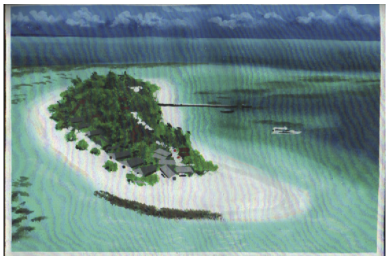
For periodic functions, the relation [link] shows that the terms $\tau\widehat{x}_k$ will attempt to approximate the Dirac-shape of X_e, meaning that the values of $\tau\widehat{x}_k$ will be huge for k/L close to a peak location but very small otherwise (see [link],

[link], and [link], compare discussion after [link]). In order to identify the amplitude X_k of the Dirac delta functions better, one uses the earlier approximation

Equation:

$$\text{periodic:} \qquad X_k \;\simeq\; \frac{1}{K}\widehat{x}_k.$$

Sampling at 300 dpi appears to be sufficient to keep the quality of the image.



Sampling at 50 dpi introduces aliasing.

Sampling of a detail at 500dpi reveals high-frequency content. These frequencies will leak from the spectral copies into the low frequencies of the main spectral copy when sampled at too low a frequency.

Aliasing

Assume that the continuous-time signal $x(t)$ is *bandlimited*, meaning there is some $B > 0$ such that $X(f) = 0$ for $|f| > B$. If the sample rate had been too small, namely

Equation:

$$f_e < 2B$$

then this copies would overlap and the original signal can not be recovered. The sampled signal shows artifacts called *aliasing* (recouvrement); these artifacts manifest is erroneous low frequency content. Such content spills or leaks from the spectral copies into the main spectral period (see [link]).

Nyquist Shannon sampling theorem:

Assume that the continuous-time signal $x(t)$ *is* bandlimited, *meaning there is some* $B > 0$ *such that* $X\left(f\right) = \quad \left\{x(t)\right\}\left(f\right) = 0$ *whenever* $|f| > B$.

If this signal x is sampled uniformly at a sufficient rate, namely at $f_e > 2B$, then $x(t)$ can be mathematically reconstructed perfectly from only those discrete samples.

Reconstruction (Ideal low pass filtering)

If the sample rate has been high enough though, namely
Equation:

$$f_e > 2B$$

then, the Fourier transform of the original signal can be recovered by cutting off the periodic copies:
Equation:

$$X\left(f\right) = X_e\left(f\right) \cdot \text{Rect}\quad \frac{f}{2f_c}$$

where f_c has to be chosen such that $B < f_c < f_e - B$ (see [link] right). (The first copy to be cut off lies over the interval $[f_e - B, f_e + B]$.) A possible choice is $f_c = f_e/2$.

In the time domain, this corresponds to convolving the sampled signal with the ideal low-pass filter (sinc). More precisely: Translating [link] into the time domain, then using the linearity of the convolution and [link], we find the ideal low-pass filtering (reconstruction) formula:
Equation:

$$\begin{aligned} x(t) &= x_e\left(t\right)*2f_c\text{sinc}\left(2f_ct\right) \\ &= \tau\sum_{n=-\infty}^{\infty} x\left(n\tau\right)\delta\left(t - n\tau\right)*2f_c\text{sinc}\left(2f_ct\right) \\ &= \left(\tau 2f_c\right)\sum_{n=-\infty}^{\infty} x\left(n\tau\right)\text{sinc}\left(2f_c\left(t - n\tau\right)\right) \end{aligned}$$

Note that the pre-factor $2\tau f_c = 2f_c/f_e$ takes into account the mismatch between sampling rate f_e and the cut-off frequency f_c. When choosing

$f_c = f_e/2$, it disappears (becomes 1).

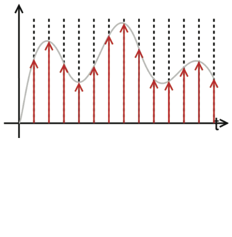
Practical reconstruction: Recall that the sinc filter is not realizable since it requires using all samples of the past and the future. Since a bandlimited signal can not be time-limited at the same time (Heisenberg), this would in theory lead to infinitely many operations.

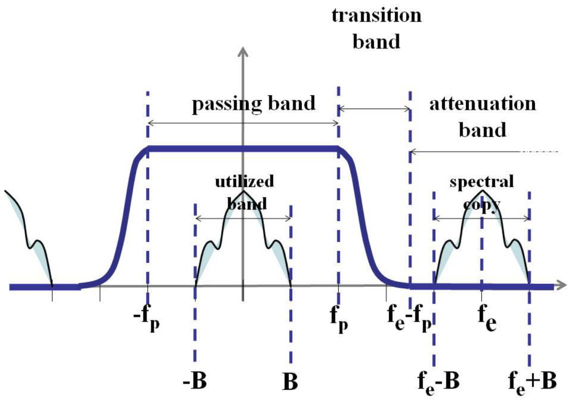
If $f_e >> 2B$, then a filter different from the ideal sinc-filter can be used. In fact, the reconstruction $x(t) = x_e(t) * b(t)$ is valid for any low-pass filter b (not only for $b(t) = 2f_c \text{sinc}(2f_c t)$) as long as we assure that the spectrum of the filter b is equal to 1 for all frequencies in $[-B, B]$ and zero for frequencies larger $f_e - B$. (Note that we do not specify the phase). Usually, one designs a filter such that

1. The filter's spectrum equals 1 in an interval $[-f_p, f_p]$ called *passing band*. It must contain the signal's frequency-band $[-B, B]$.
2. The filter *attenuates* all frequencies beyond $f_e - f_p$.
3. The *transition band* of the filter is $[f_p, f_e - f_p]$. This range of frequencies can be chosen as large as $[B, f_e - B]$ by setting $f_p = B$ and as short as desired, but always containing the point $f_e/2$ by setting f_p smaller but as close to $f_e/2$ as desired.

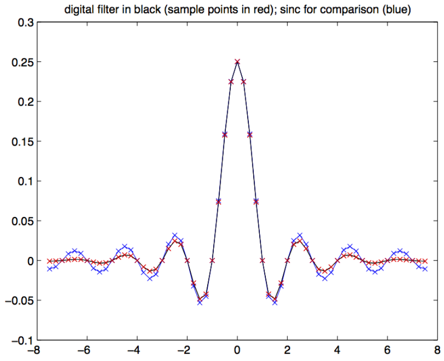
The Dirac Comb modulated by a signal.

The spectrum of the sampled signal (note the spectral copies) and a practical reconstruction filter: It's passing band contains the band $[-B, B]$ of the signal and it attenuates all spectral copies.

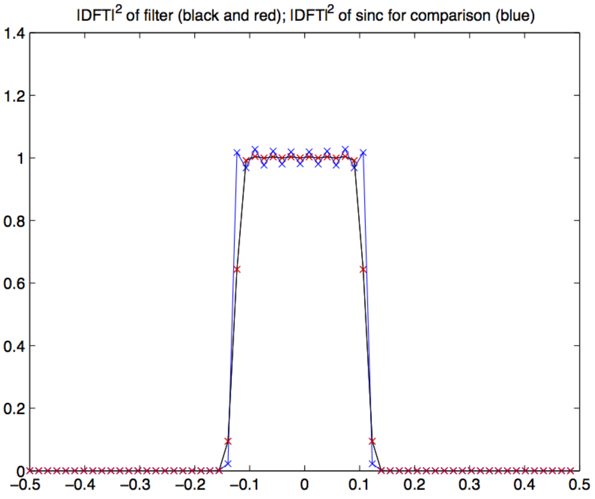




The low-pass FIR filter of length 61 and ideal cutoff at $f_c = 1/8 = 0.125$ as produced by Matlab. The filter values are in red. For comparison in blue the corresponding samples of the ideal filter $2f_c\text{sinc}\left(2f_ct\right)$ at $t = -30, .., -1, 0, 1, ..30.$



DFT of the signals on the left. The actual transition band of the filter is $[0.1, 0.15]$. Note that the sinc-samples provide a filter with sharper cutoff but slightly lower quality in the pass-band since it requires infinitely many samples. The power computed from either side (recall [link]) is $P_b = 0.003'891 = 0.237'4/61 \simeq 2f_c/$ length.



All this assures that we keep the entire spectral information of $x(t)$ contained in $[-B, B]$, yet remove all spectral copies (see [link]). Sampling an ideal reconstruction filter is a method to a design a low pass filter. Here, point 2 in the above list helps avoiding aliasing in such a filter. However, the resulting filters are not ideal as FIR filters possible since not all samples can be used, resulting in Gibbs-like phenomena (see [link] on the right).

Matlab uses the command `fir1` to compute finite impulse response filters (FIR). A larger transition band allows for shorter filters to be used, thus speeding up computation. [link] shows a length 61 FIR filter with ideal cutoff frequency at $f_c = 1/8 = 0.125$. Its transition band is roughly $[0.10, 0.15]$. As is common in digital filtering, the sampling frequency is assumed to be $f_e = 1$.

Power and Energy from Sampling

We summarize a few facts.

Good quality when above Nyquist We are now in a position to make more precise how many samples are enough to well approximate the power of a signal via [link], or its energy per time over a large interval via [link], depending on the context. Roughly, the signal must be sampled at least at Nyquist rate so that the samples and their DFT faithfully represent the signal and its Fourier transform. Clearly, sampling at a much higher rate than Nyquist will improve the approximation.

Power independent of sampling rate We conclude that changing the sampling rate will not change power, as long as we stay above Nyquist, and at least approximately. Before we study how to change the sampling rate in the next sections, we give a quick demonstration of [link] using a simple band-limited signal: a filter.

Ideal filter and FIR filter The energy of the ideal filter with cutoff frequency f_c is $||2f_c\text{sinc}\,(2f_ct)||^2 = 2 \cdot f_c$ which follows easily from the power spectrum being 1 on an interval of length $2 \cdot f_c$ and zero else, using Plancherel. To compute the power of a *digital low pass FIR filter*$b = (b_0, ..., b_{K-1})$ of length K with approximate cutoff frequency f_c we study its DFT $\hat{b}$: we note that $\hat{b}_k = 1$ for roughly $2f_cK$ of the indices k and $\hat{b}_k = 0$ for the other indices. Note that $f_e = 1$ and $0 < f_c < 1/2$ for a digital filter. Using [link] we find
Equation:

$$P_b = \frac{1}{K^2} \sum_{k=0}^{K-1} \left| \hat{b}_k \right|^2 \simeq \frac{1}{K^2} 2 f_c K = 2 f_c / K = 2 f_c / L.$$

This fits perfectly with [link] since for a digital filter $f_e = 1$ and, thus, $K = L$. The approximation becomes better, the sharper the transition band of the filter b , i.e., the longer the filter is.

Compare again with [link] for a concrete filter with length $K = 61$, $f_c = 1/8 = 0.125$ and power $P_b = 0.003'891$.

Decimation and Downsampling

Pure Downsampling

The easiest case consists in reducing the sampling rate by simply dropping samples. This procedure is called *Downsampling*. However, we need to be careful on the effect of this procedure. We note immediately, that the new sampling rate $f_d = f_e / M$ needs to be still above Nyquist, i.e., $f_e > M \cdot 2B$ in order to avoid aliasing.

- **Downsampling** Assume that $f_e > M \cdot 2B$.
  Equation:

$$(y_0, y_1, y_2, \dots) \rightarrow \ \downarrow M \ \rightarrow (y_0, y_M, y_{2M}, \dots) = (z_0, z_1, z_2, \dots)$$

Effect in the frequency domain In continuous time, downsampling the signal $y(t)$ corresponds to passing to the signal $z(t) = y(Mt)$. Indeed, sampling z at the same frequency as y we obtain the samples $z_k = y_{kM}$ as we should. In summary, using the properties of the Fourier transform
Equation:

$$z\left(t\right) = y\left(Mt\right) \qquad Z\left(f\right) = \frac{1}{M} Y\left(\frac{f}{M}\right)$$

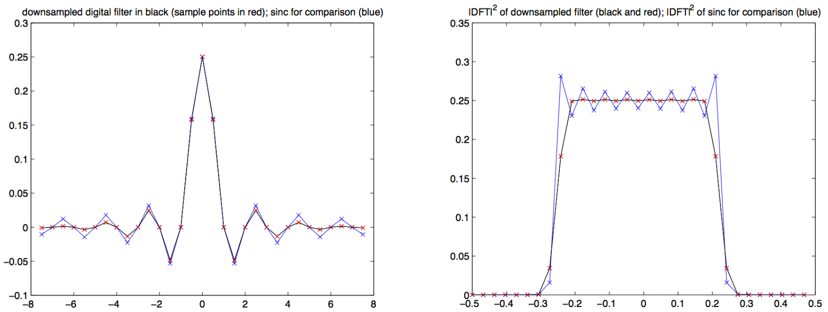
This tells us, that the spectral copies of Z_e (the Fourier transform of the sampled signal $z_e\left(t\right)$) should have the same overall shape and at the same distance f_e as those of Y_e, but they are stretched in frequency by a factor M and squeezed in amplitude by a factor M.

Indeed, we arrive at the same conclusion computing via the sampled signals, using [link]:
Equation:

$$Z_e\left(f\right) = \sum_k \tau z_k e^{-j2\pi f k\tau} = \sum_k \tau y_{kM} e^{-j2\pi f k\tau} = \frac{1}{M} \sum_k (M\tau) y_{kM} e^{-j2\pi (f/M) k (M\tau)} = \frac{1}{M} Y_d\left(\frac{f}{M}\right)$$

Recall that $Y_d\left(f\right)$, the Fourier transform of y sampled at frequency $f_d = f_e / M$, looks like Y_e except that its spectral copies lie at distance f_d, thus M times closer than those of Y_e.

For the *discrete Fourier Transform* we should expect to see roughly the same behavior. Recall, though, that the relation between continuous and discrete Fourier transform is only approximative.



The result of downsampling the filter from [link] by a factor $M = 2$ on the left, and its power spectrum $|\text{DFT}|^2$ on the right (black and red). Again, the ideal sinc filter in blue for comparison. Note that the filter's cutoff frequency has increased by M (stretching of the spectrum by M) and that its $|\text{DFT}|^2$ values have decreased by $1/M^2 = 0.25$ (see text). The power computed from either side is 0.003829 (recall [link]): downsampling of a signal with $2B < f_e/M$ does not change power.

Aliasing: We give two examples.

First, downsampling to a sampling rate that is too low can lead to aliasing. Downsampling the image of [link] left leads to the same effect as visible in the same figure center.

Second, downsampling a simple signal such as the filter b of [link] by $M = 2$ results in a new filter b' with twice the cutoff frequency, i.e. $f_c' = 1/4 = 0.25$, but with a value $\hat{b}_k = 1/2 = 1/M$ over the pass-band, thus a power spectral value $1/4 = 1/M^2$ over the pass-band (see [link]). No aliasing occurs since the condition $f_e > M \cdot 2B$ is satisfied.

Power: We consider first the case of a T-periodic signal y. Then, $z(t) = y(Mt)$ has period T/M. Substituting $s = tM$ with $ds = Mdt$ we get
Equation:

$$P_z \quad = \quad \frac{1}{T/M}\int_0^{T/M} |z(t)|^2 dt = \frac{M}{T}\int_0^{T/M} |y(Mt)|^2 dt = \frac{1}{T}\int_0^{T} |y(s)|^2 ds = P_y$$

From the simple properties we know $Z_k|_{f=\frac{k}{T/M}} = Y_k|_{f=\frac{k}{T}}$ (same complex amplitudes[footnote], but belonging to different frequencies). Using Parseval we see again that the power does not change. See properties of Fourier Series.
There is no contradiction between $Z_k = Y_k$ and $Z(f) = \frac{1}{M}Y\left(\frac{f}{M}\right)$. Both express that $z(t) = y(Mt)$, both allow to conclude that power does not change under downsampling, and both imply that the DFT of z is M times smaller and corresponds to frequencies which are M times further apart than those of y. Indeed, using [link] we get
Equation:

$$\hat{z}_k\Big|_{f=\frac{k}{T/M}} = \frac{K}{M} Z_k = \frac{1}{M} K Y_k = \frac{1}{M}\hat{y}_k\Big|_{f=\frac{k}{T}}$$

and using [link] with L equal to the correct period and number of samples (T/M and K/M for $z(t)$ respectively T and K for $y(t)$) we get again
Equation:

$$\hat{z}_k\Big|_{f=\frac{k}{T/M}} = \frac{K/M}{T/M} Z\left(\frac{k}{T/M}\right) = \frac{K}{T}\frac{1}{M} Y\left(\frac{1}{M}\frac{k}{T/M}\right) = \frac{1}{M}\frac{K}{T} Y\left(\frac{k}{T}\right) = \frac{1}{M}\hat{y}_k\Big|_{f=\frac{k}{T}}$$

Note that the energy of a finite energy signal would change under downsampling: computing in time
Equation:

$$||z||^2 \;=\; \int_{-\infty}^{\infty} |z(t)|^2 dt = \int_{-\infty}^{\infty} |y(Mt)|^2 dt = \frac{1}{M}\int_{-\infty}^{\infty} |y(s)|^2 ds = \frac{1}{M}||y||^2$$

A simliar computation can be done in frequency. See Comment 6.

Comment 6 Computing in frequency with $f = Mg$ and $df = Mdg$:
Equation:

$$||Z||^2 \;=\; \int_{-\infty}^{\infty} |Z(f)|^2 df = \int_{-\infty}^{\infty} \left|\frac{1}{M} Y\left(\frac{f}{M}\right)\right|^2 df = \frac{1}{M}\int_{-\infty}^{\infty} |Y(g)|^2 dg = \frac{1}{M}||Y||^2$$

For a discrete signal we are naturally in the periodic case. From the above we should expect that power stays approximatively the same under downsampling provided that $f_e > M \cdot 2B$. Also, we noted earlier that power should not depend on the sampling rate, as long as the samples faithfully represent the signal, and at least approximatively.

For the simple signal b, the filter from [link], we may verify this explicitly. Denote the downsampled filter by b' (see [link] for an illustration with $M = 2$). Since no aliasing occurs during downsampling, the pass-band is now M times longer, meaning that M times more of the $\widehat{b'}_k$ are different from zero (this makes the power increase by M). Further, their power spectral values $\left|\widehat{b'}_k\right|^2$ are by M^2 smaller (this makes the power decrease by M^2). Finally, the sample length is now M times shorter (this increases the power my M; recall [link]).

All in all, power is not changed, at least approximatively. One finds $P_{b'} = 0.003'829$ which has to be compared to the power $P_b = 0.003'891$ of the original filter.

To obtain a low-pass filter one would have to normalize b' to $b'' = M \cdot b'$.

Decimation

Let us now drop the assumption $f_e > M \cdot 2B$.

To resample a signal $x(t)$ at an M times lower rate, a first attempt would be to discard all but every M-th sample: $z_k = x_{kM} = x\left(kM\tau\right)$. This step is called *downsampling* as we have seen above. However, to avoid aliasing effects caused by downsampling below Nyquist rate a low-pass filtering at cutoff $B^* = f_e / \left(2M\right)$ is required *before* downsamling. The filter used is called *anti-aliasing filter*. The *new utilized bandwidth* will be only B^*.

The procedure of first applying an anti-aliasing filter and then downsampling is called *decimation*.

Agreeably, the filtering before downsampling destroys information about $x(t)$. However, this loss occurs in a controllable manner: It removes high-frequency information. For an audio signal, we loose the high pitch sound. For an image we loose sharpness of edges. This should be compared to an uncontrolled loss of quality when no anti-aliasing filter is applied (see [link]).

In summary, **Decimation by** M means to resample at M times lower rate $f_d = f_e/M$ and consists of two steps:

1. **low-pass filtering** the samples at cutoff frequency $\frac{1}{2M}$
   Equation:

$$(x_0, x_1, x_2, \ldots) \rightarrow \quad \cap\frac{1}{2M} \quad \rightarrow (y_0, y_1, y_2, \ldots)$$

2. **Down-sampling** (in French: decimation)
   Equation:

$$(y_0, y_1, y_2, \ldots) \rightarrow \quad \downarrow M \quad \rightarrow (y_0, y_M, y_{2M}, \ldots) = (z_0, z_1, z_2, \ldots)$$

Low-pass filtering will reduce power, as high frequencies are attenuated. Down-sampling will leave the new power roughly the same.

The matlab commands are `decimate` and `downsample`.

Interpolation and Upsampling Interpolation

Let us now look at increasing the sample rate. Since it is less obvious how to achieve this, let us first consult theory.

Given are the samples x_n x $n\tau$ of a band-limited signal x t taken at frequency f_e τ is above the Nyquist rate B. We want to compute the samples z_k of x t at a higher sampling rate f_u N f_e. This means that the new sampling step should be f_u f_eN τ N or, z_k x $k\tau$ N .

Since the original sampling rate f_e B is above Nyquist, we can in theory reconstruct the entire signal x t using the reconstruction formula [link] using f_c f_e . Once the continuous-time (finite energy) signal x t is obtained, we only need to sample it at t $k\tau$ N k f_u. This reads as follows
Equation:

$$z_k \qquad z\; k\tau \qquad x\; k\frac{\tau}{N} \qquad {}_{n} \qquad x_n \qquad \frac{k\frac{\tau}{N} \qquad n\tau}{\tau} \qquad {}_{n} \qquad x_n \qquad \frac{k \quad nN}{N}$$

This formula allows indeed to compute z_k from x_n, at least in principle. However, a closer look at theory is required to understand the effect when using only finite many samples. The reconstruction formula [link] is best understood in the frequency domain: it amounts to removing the spectral copies of X_e f via filtering with cut-off frequency f_c f_e . To make this filtering step visible we need to write [link] in form of a convolution. To this end, we write
Equation:

$$z_k \qquad {}_{m} \qquad y_m \qquad \frac{k \quad m}{N} \qquad y_m \qquad \frac{m}{N} \qquad N\; y_m \qquad \frac{}{N} \qquad \frac{m}{N}$$

where the new sequence y_m is obtained by “upsampling” and is given as:
Equation:

$$y_m \qquad \begin{matrix} x_n & m\; N & n \\ & m\; N & \end{matrix}$$

The convolution [link] allows for more convenient data processing via *digital filtering* and for a simple spectral interpretation.

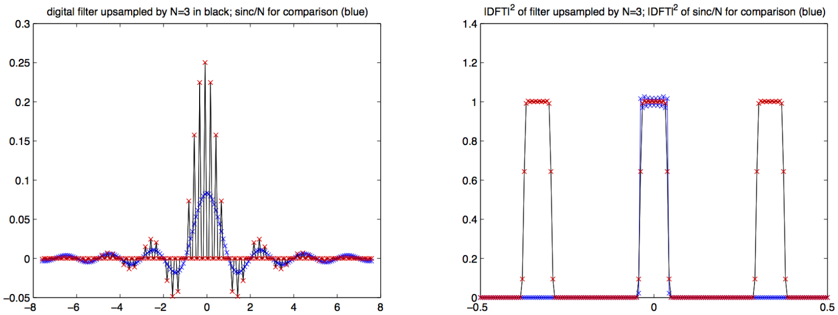
Interpolation by N or resampling at N times larger rate consists of the following steps:

1. **Up-sampling** (in French: “interpolation”)
   Equation:

$$x \quad x \quad x \qquad N \quad x \quad {}_{N} \quad x \quad {}_{N} \quad x \quad {}_{N} \quad x \qquad y \quad y \quad y$$

2. **Multiplication** by N and **Low-pass filtering** at cutoff frequency $\frac{}{N}$ using the ideal filter $\frac{}{N}$ $\frac{m}{N}$:
   Equation:

$$y \quad y \quad y \qquad N \quad \frac{}{N} \quad y_m \quad \frac{m}{N} \quad z \quad z \quad z$$



The result of upsampling the filter from [link] by a factor N on the left, and its power spectrum on the right (black and red). Again, the ideal sinc filter (divided by N to adjust for the zero-samples) in blue for comparison. Note that the filter's cutoff frequency has decreased by N (contraction of the spectrum by N) and that its values have not changed (see text). The power computed from either side is 0.001'297 and has decreased by a factor N from the original power of 0.003'829: only upsampling of a signal with B f_e decreases power by the upsampling factor.

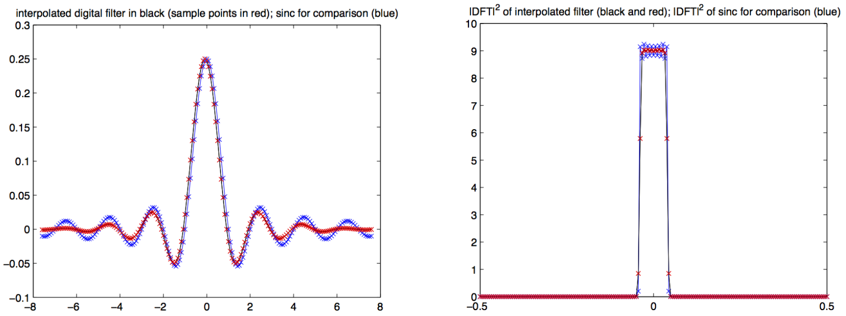
Spectral picture of Interpolation by N

In analogy to the decimation we set $z\ t$ $x\ t\ N$. Then, the samples of $z\ t$ taken at the same rate f_e constitute the samples of x taken at rate f_u. Analogously to the decimation we find quickly
Equation:

$$z\ t \qquad x\ t\ N \qquad Z\ f \qquad NX\ fN$$

which indicates how the spectrum at rate f_u Nf_e is obtained: the spectrum at rate f_e gets contracted in the frequency axis by N and expanded in amplitude by N.

Note that the spectral copies of Z_e are at distance f_e just like those of X_e.



The result of interpolating (upsampling and filtering) the filter from [link] by a factor N on the left, and its power spectrum on the right (black and red). Again, the ideal sinc filter (no need to divide by N since the zero-samples are now corrected through the filtering step) in blue for comparison. Note that the fundamental period contains now only one spectral copy, as it should, as a result from the filtering. The spectrum is contracted by N in frequency and expanded in amplitude by N . The power computed from either side is 0.003'891 and almost identical to the original power of 0.003'829: interpolation of a signal with B f_e does not change power.

We may break the procedure down into the individual steps:

(i) Upsampling (introducing the zero-samples) leaves the Fourier transform, and thus the spectrum almost intact, leading only to a rescaling of the frequencies (contraction of X): Indeed, the Fourier transform Y_e of the samples y_m becomes

Equation:

$$Y_e \quad f \qquad \underset{m}{} \tau y_m e^{j\ \pi f m \tau} \qquad \underset{n}{} \tau x_n e^{j\ \pi f n N \tau} \qquad X_e \quad Nf$$

The corresponding signal $y\ t$ (with samples y_m at sampling rate f_e) is not of interest. If you want to know about it any way, see Comment 7. Note, however, that the fundamental period of Y_e amounts to f_e and contains N copies of $X\ Nf$ (see [link]).

Comment 7 For clarity: the Fourier transform of y is found by removing the spectral copies of Y_e *outside* f_e f_e . These copies are caused by sampling. The Fourier transform of y consists of N

contracted copies of X at distance f_e N of each other which are caused by upsampling. Using the reconstruction formula [link] with the samples y_m, sample rate f_e and correct pass-band f_e yields the signal y t y_m $\frac{t \ \ m\tau}{\tau}$. Using that while m for all integer m we find quickly that the samples of y t are indeed y $k\tau$ y_k, i.e., x x x .

(ii) Multiplication with N restores the average value of the samples. The Fourier transform is now NX_e Nf which consists of copies of NX fN , or Z f , at distance f_e N (as for Y_e, there are N copies in one period).

(iii) The digital low-pass filtering of Ny_m at cut-off frequency N removes all of the copies of NX fN except the ones centered at 0, f_e, f_e etc. and leaves only one copy per period, in other words, only copies at distance f_e. What remains is exactly Z_e f as we have pointed out earlier. (see [link]).

Power and Interpolation

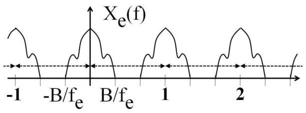
Similarly as with the decimation the power of a periodic signal does not change under interpolation. In fact, we only need to use Z f NX fN and replace M by N in the computation done with the decimation. We conclude that *the power of discrete samples does not change under interpolation* provided that f_e B, at least approximatively.

To verify this, let us move through the 3 steps above. Step (i), upsampling, reduces power by a factor N since the sum of squares of the samples is the same (the zeros added don't contribute), but there are now N times more samples. Power is an average.

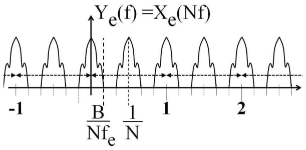
Step (ii) obviously multiplies power with N . Step (iii), the low-pass filtering, removes N spectral copies and leaves only 1, thus divides power by N. All steps together leave the power as it is.

Interpolation

Interpolation illustrated in the Spectral Domain with N .
Horizontal arrows indicate spectral copies. To keep the sample rate at the same value f_e one changes from X_u, resp. Y_d to Z_e (see d).

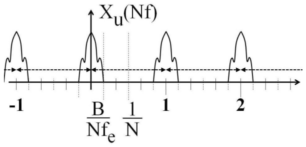


N
signal is correctly sampled (
B f_e)



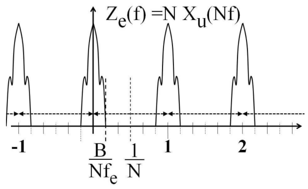
$\frac{}{N}$

inserting N zeroes contracts the Fourier transform by a factor N

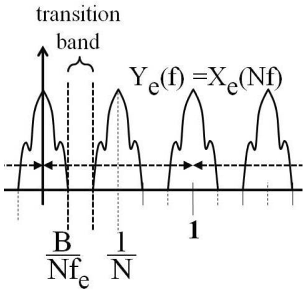


N

digital low-pass filtering with cut-off frequency at N leaves only one of the contracted copies per period 1, i.e., it leaves X_u Nf



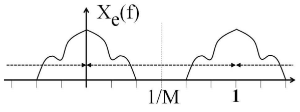
If B $\frac{f_e}{}$ multiplying with N leads to $N X_u$ $N f$ Z_e f where z t x t N . Sampling z at rate f_e provides the desired samples at rate f_u of x.



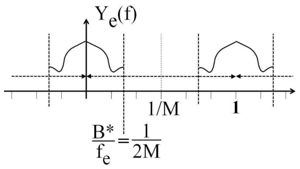
If B f_e , then a transition band
$\frac{B}{Nf_e}$ $\frac{}{N}$ $\frac{B}{Nf_e}$
is feasible

Decimation

Decimation illustrated in the Spectral Domain with M .
Horizontal arrows indicate spectral copies. To keep the sample rate at the same value f_e one changes from Y_d to Z_e (see d).

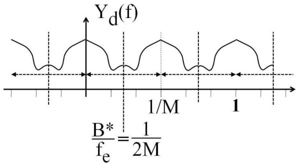


$\frac{}{M}$
M marks the center of the first spectral copy after decimation

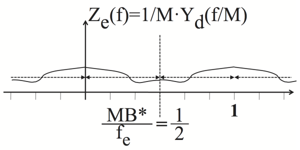


M
digital low-pass filtering with (ideal) cut-off frequency at
M ensures no aliasing after decimation; the *new utilized*

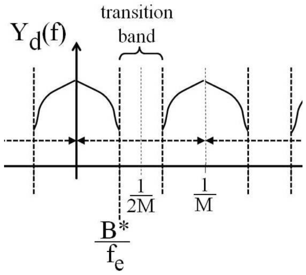
bandwidth is B f_e M ; high frequency information might be lost, resulting in a new signal y.



adjusting sampling rate After decimation only the samples y_{kM} remain, corresponding to a sampling of y at f_d f_e M, with Fourier transform Y_d f .



If B $\frac{f_e}{M}$ The samples z_k y_{kM} correspond to z t y Mt sampled at f_e, and Z_e f M Y_d f M (see text).



Choosing the *new utilized bandwidth* to be $B \qquad \frac{f_e}{M}$, then a transition band $\frac{B}{f_e} \quad \frac{}{M} \qquad \frac{B}{f_e}$ is feasible

Sampling Rate Conversion

Sampling rate conversion

In general, sampling rates are converted only in rational fractions such as from 32 to 48 kHz by a factor of 3/2 which are performed by appropriate sequences of up- and down-sampling.

Practical considerations:

- First interpolate, then decimate (preserve signal quality)
- the low-pass filters used after upsampling (2nd step of interpolation) and before downsampling (1st step of decimation) can be combined into the one filter with the smaller pass-band.
- Interpolation and Decimation by large factors should be done in steps, keeping N and M below or at 6 at most.
- Matlab commands: `decimate`, `downsample`, `interp`, `upsample`, `resample`. See the corresponding help manual (`help decimate` etc).
- The ideal sinc filter is not realizable since it requires knowledge of the infinite past and future; one uses FIR filters with linear phase to avoid audible artifacts.

Models of Noise

White Noise In a nutshell, a sequence $(\varepsilon_1, \varepsilon_2, \varepsilon_3, \dots)$ is called *white noise* if its entries ε_k are *independent* identically distributed random numbers.

- “Independence” means that there is no information or relation between the members ε_k; in other words, knowing or observing the sequence until time t, i.e., $(\varepsilon_1, \dots, \varepsilon_t)$ does not allow to predict the next member ε_{t+1} any better than if nothing had been observed.
- “Random” means that in principle, an entry can take any value in some given set with certain chance, and that it is are not in advance which value it will take.
- “Identically distributed” means that each entry has the same chances to assume a possible value.

White noise is an example of an *stationary, ergodic* series. *Stationarity* means that the statistics don't change over time. *Ergodicity* means that statistical measures such as mean and variance can be estimated by observing enough samples of one *single* sequence $(\varepsilon_1, \varepsilon_2, \varepsilon_3, \dots)$; the mean or expectation[footnote] of an entry $\mathbb{E}[\varepsilon_1]$ can be estimated as the sample mean $(1/N)\,(\varepsilon_1 + \dots + \varepsilon_N)$.
Recall that $\mathbb{E}[X]$ denotes the expectation of the random variable X.

An example of a stationary, *non ergodic* series is the one where ε_1 equals 1 or -1 with equal probability, and all other ε_k are equal to ε_1. Clearly, the sample mean of *one* single sequence is then either 1 or -1, but not 0 as it should.

An example of white noise is the error ε_k introduced by quantization, i.e., $\varepsilon_k = y_k - x_k$ where x_k is a “typical” signal before and y_k the signal after quantization. Check of Randomness and Independence: as we are observing the first t samples $(\varepsilon_1, \dots, \varepsilon_t)$ we have no indication whatsoever on the quantization error of the $t+1$st sample — unless the signal is very special. [An example of an atypical signal would be one that is already quantized: after observing 500 times an error 0 we start to suspect that the future errors will also be 0.] Check of the distribution: a quantization done by rounding to the third decimal, e.g., will result in errors that lie between $[-0.00049999\dots, 0.0005]$ where all values in this interval are equally likely. This means, e.g., that ε_k is negative with chance 1/2 and that, e.g., ε_k is within $[0.0002, 0.0003]$ with chance $1/10$. Since this is the same of all entries ε_k, they are identically distributed.

Spectral analysis of stationary signals and series

By their own nature, similarly to periodic signals, stationary signals and series have no finite energy. As a simple example consider the sequence $\varepsilon_k = \pm 1$ with random sign. Since $\varepsilon_k^2 = 1$ for all k, the energy of the sequence is clearly infinite. In order to arrive at a meaningful spectral analysis one defines the *power* of a stationary signal $x(t)$ as the time average of the energy:
Equation:

$$P := \lim_{L\to\infty} \frac{1}{L} \int_{-L/2}^{L/2} |x(t)|^2 dt$$

Note that for a periodic signal definition [link] gives the same value as [link].

While periodic signals possess a natural Fourier expansion into a series, we need to take time-averaged, windowed Fourier transforms for stationary signals; also, due to randomness, one needs to take averages over different realizations in order to obtain a deterministic non-random spectral descriptor. Most useful is the *power spectrum*$S(f)$ which corresponds to the square of the absolute value of the time-averaged windowed Fourier transform:
Equation:

$$S\left(f\right) := \lim_{L\to\infty} \frac{1}{2L+1}\mathbb{E}\left|\sum_{k=-L}^{L}\varepsilon_k e^{-j2\pi fk}\right|^2 \overset{\text{station.}}{=} \lim_{L\to\infty}\frac{1}{2L+1}\mathbb{E}\left|\sum_{k=1}^{2L+1}\varepsilon_k e^{-j2\pi fk}\right|^2$$

Sometimes it is (erroneously) written as $S\left(f\right) = \left|E\left(f\right)\right|^2$. erroneously because it should be averaged: $S\left(f\right) = \mathbb{E}\left|E\left(f\right)\right|^2$. The famous Wiener-Khinchine theorem says, that
Equation:

$$S\left(f\right) = \sum_{-\infty}^{\infty} r\left(k\right)e^{-j2\pi fk} = \mathscr{F}\{r(k)\}\left(f\right)$$

where $r\left(k\right) = \mathbb{E}\left[\varepsilon_n \cdot \varepsilon_{n-k}\right] = \mathbb{E}\left[\varepsilon_k \cdot \varepsilon_0\right]$ is the *auto-correlation* of the process. Note $S(f) = S(f+1)$.

Connection to Discrete Fourier transform (DFT) In practice, a finite length signal $\varepsilon_0, \ldots, \varepsilon_{K-1}$ is interpreted as periodically repeated (with period K). Its Fourier transform is then a periodic series again, called DFT, which can be computed via an algorithm called FFT in only $K\ \log\ (K)$ computations according to the formula
Equation:

$$\text{DFT:} \qquad \hat{\varepsilon}_n = \sum_{k=0}^{K-1}\varepsilon_k e^{-j2\pi kn/K}$$

Recall that Matlab starts indices always with 1, thus interpreting the first entry of the vector $\hat{\varepsilon}$ as corresponding to frequency 0 (see matlab `help fft`).

If we set $K = 2L+1$ we find that $\hat{\varepsilon}_n$ appears on the right side of [link]. If we assume that L is large so that we can neglect the limit, we find that $S(f)$ indicates what to expect of the squared FFT coefficients (i.e., the FFT-power-spectrum) on average:
Equation:

$$\mathbb{E}\left|\hat{\varepsilon}_n\right|^2 \simeq K \cdot S\left(\frac{n}{K}\right)$$

for $n = 1, .., K$. Vice versa, the samples of the power spectrum $S(f)$ for $f = k/K$ can be estimated from averaging the FFT coefficients over several noise signals each of the form $\varepsilon_1, \ldots, \varepsilon_{2L+1}$:
Equation:

$$S\left(\frac{k}{K}\right) \simeq \frac{1}{K}\mathbb{E}\left|\hat{\varepsilon}_k\right|^2$$

for $k = -L, \ldots, 0, \ldots, L$ (it is more natural to represent S over a symmetric interval; recall that $\hat{\varepsilon}_{-L} = \hat{\varepsilon}_{L+1}, \ldots, \hat{\varepsilon}_0 = \hat{\varepsilon}_{2L+1}$).

Flat spectrum of zero mean white noise

The power spectrum $S(f)$ of white noise with zero mean ($\mathbb{E}[e_k] = 0$) can be computed as follows. First we find
Equation:

$$r(0) = \mathbb{E}\left[\varepsilon_0^2\right] = \sigma^2 \quad \text{and} \quad r_k = \mathbb{E}\left[\varepsilon_k\right] \cdot \mathbb{E}\left[\varepsilon_0\right] = 0 \quad k \neq 0$$

by independence. By [link] we find
Equation:

$$S(f) = \sigma^2 = \text{Power}$$

for all f. Note that the power spectrum of white noise is *flat*. Its constant value is equal to the variance of the zero mean noise. The constant value must clearly be equal to the power. (Also: $P_\varepsilon = 1/K \sum \varepsilon_n^2 \simeq \mathbb{E}\left[\varepsilon_0^2\right] = \sigma^2$ from statistics.)

Consequently, since the *mean* of the DFT, i.e. $\mathbb{E}|\hat{\varepsilon}_n|^2$ is constant, the average of the DFT when computed over many realizations of the noise is nearly flat[footnote]; however, the FFT of any single realization will show large oscillations (see [link] 3rd).
matlab demo: plot(mean(abs(fft(0.1*(rand(1000,1000)-0.5),[ ],2).^2)));axis([0 1000 0 0.01])

Nevertheless, the power can be estimated from one realization: using [link], using ergodicity to estimate the expected value $\mathbb{E}[\dots]$ simply as the average over all samples, and using [link] we get
Equation:

$$S\left(\frac{k}{K}\right) \simeq \frac{1}{K}\mathbb{E}|\hat{\varepsilon}_k|^2 \simeq \frac{1}{K} \cdot \frac{1}{K}\sum_{n=0}^{K-1} |\hat{\varepsilon}_n|^2 = P_\varepsilon.$$

This computation also confirms that the spectrum of white noise is flat and that its constant value is equal to the power.

In fact, a direct computation from [link] shows that the FFT output $\hat{\varepsilon}_n$ is a white noise as well, however with variance $K\sigma^2$ and so [link] (the first approximation in [link]) holds for white noise exactly, not only approximatively.

Intuitively, the formula $S(f) = \sigma^2$ means that all frequencies are present in white noise with equal overall-amplitude. The spectrum being flat is a direct consequence of the *independence* between the noise terms. We should recall, though, that strictly speaking the spectrum is flat only as an average over many noise realizations.

Application: Quantization In many application, such as inference in wireless communication and such as in quantization in signal processing, a commonly accepted model of the effect of a source of errors on the signal $x(t)$ is the so-called *additive white noise model*:
Equation:

$$x_k \rightarrow +\varepsilon_k \rightarrow y_k$$

For a quantization with precision Δ the error is the quantification error $\varepsilon_k = y_k - x_k$ and it is determined by the signal itself. [One could argue, that ε_k is not random since it is completely determined once x_k is known. Still, the model is useful since we can usually not predict ε_{k+1} from observing $x_1, \dots, x_k$.]

For quantization using rounding (matlab: `round`[footnote]) it can be shown that the noise power per sample amounts to $P_{\text{Quant.noise}} = \sigma^2 = \frac{\Delta^2}{12}$. When using one more bit for quantization, then the error Δ is half as large, thus the power 4 times smaller, which amounts to roughly -6 dB. In other words, the power of the quantization noise is proportional to -6dB times the number of bits used.

The error ε_k is in this case uniformly distributed on the interval $[-\Delta/2, \Delta/2]$.

Using the analog of Parseval's equation and recalling that $S(f)$ is the power spectrum (analog of the square of the Fourier transform) we have indeed:
Equation:

$$S_e\left(f\right) = \frac{\Delta^2}{12}\mathrm{Rect}\left(\frac{f}{f_e}\right) \qquad P_{\text{Quant.noise}} = \frac{1}{f_e}\int_{-f_e/2}^{f_e/2} S_e\left(f\right)df = \frac{\Delta^2}{12}$$

For K samples of noise taken over a time interval of length K/f_e we have, thus, approximatively
Equation:

$$\frac{1}{K}\sum_{k=0}^{K-1}\left|\varepsilon_k\right|^2 = \frac{1}{K^2}\sum_{k=0}^{K-1}\left|\hat{\varepsilon}_k\right|^2 \simeq P_{\text{Quant.noise}} = \frac{\Delta^2}{12}$$

Note that the FFT increases power by K. The relation [link] becomes exact when taking expected values $\mathbb{E}$. The approximation improves the larger K is, since the left side is an estimator of the variance σ^2, which is $\Delta^2/12$ for quantization with precision Δ.

Application: Interference A further example of a situation where an additive white noise proves useful is wireless transmission of a binary signal under interference. Here, bits may flip from 0 to 1 and vice versa since detection is not perfect. The error $\varepsilon_k = y_k - x_k$ is here determined by the interfering signal, and the configuration of the decoder. Note that the possible values of each ε_k is either 0 (no flip) or 1 (flip). Without specific information on the interference, the chance of a flip is independent of the time k, and independent of the past occurrence of flips. Thus, white noise is a very reasonable model for ε_k. Clearly, the probability $P[\varepsilon_k = 0]$ will be close to 1 if only little inference is present and will decrease the stronger the inference.

Gaussian noise As an important special case we mention Gaussian white noise, where the common distribution of the ε_k is Gaussian, or “normal”: This model assumption is standard whenever nothing is known on the distribution. It makes sense, e.g., as model for an overall error which is composed of several small unknown errors. (compare Central Limit Theorem)

Colored Noise A sequence $(n_1, n_2, n_3, \dots)$ is called *colored noise* if its terms n_k are random and possess a relation or dependence between them. Consequently, the power spectrum of colored noise is not flat, but possesses certain prevalent frequencies — hence the name “colored” (recall that the frequencies of light waves correspond to colors).

One of the most simple ways to produce colored noise is to filter white noise. For instance, $m_k = \varepsilon_k + \varepsilon_{k+1}$ is colored since the entries m_k are no longer independent: $m_0 = \varepsilon_0 + \varepsilon_1$ and $m_1 = \varepsilon_1 + \varepsilon_2$ contain the same number ε_1 as an additive term. Similarly, $n_k = \varepsilon_k - \varepsilon_{k+1}$ is colored.

Adopting a continuous-time notation (for convenience) we write
Equation:

$$m(t) = \varepsilon(t) + \varepsilon(t+1) \qquad n(t) = \varepsilon(t) - \varepsilon(t+1)$$

with Fourier transform
Equation:

$$\begin{aligned} M(f) &= E(f) + E(f)e^{-j2\pi f} \\ &= E(f)\left(1 + e^{-j2\pi f}\right) \\ &= E(f)\left(e^{j\pi f} + e^{-j\pi f}\right)e^{-j\pi f} \\ &= E(f)2\cos(\pi f)e^{-j\pi f} \end{aligned}$$

and similarly

Equation:

$$N(f) = E(f) - E(f)e^{-j2\pi f} = E(f)\left(1 - e^{-j2\pi f}\right) = E(f)\left(e^{j\pi f} - e^{-j\pi f}\right)e^{-j\pi f} = E(f)2j\sin(\pi f)e^{-j\pi f}$$

and power spectrum[footnote], (use that $2\cos^2(x) = 1 + \cos(2x)$) and $2\sin^2(x) = 1 - \cos(2x)$)

The same formula for $|M(f)|^2$ and $|N(f)|^2$ could be obtained by computing it as the Fourier transform of the auto-correlation (see [link]); indeed, for $m(t)$: $r(0) = 2\sigma^2$, $r(1) = r(-1) = \sigma^2$ and $r(k) = 0$ for all other k; for $n(t)$ the same except $r(1) = r(-1) = -\sigma^2$.

Equation:

$$|M(f)|^2 = |E(f)|^2 4\cos^2(\pi f) = |E(f)|^2 2(1 + \cos(2\pi f)) = 2\sigma^2(1 + \cos(2\pi f))\mathrm{Rect}(f)$$

Equation:

$$|N(f)|^2 = |E(f)|^2 4\sin^2(\pi f) = |E(f)|^2 2(1 - \cos(2\pi f)) = 2\sigma^2(1 - \cos(2\pi f))\mathrm{Rect}(f)$$

where σ^2 is the total power of the original noise ε_k and $S(f) = |E(f)|^2 = \sigma^2$. Note that neither $|M(f)|^2$ nor $|N(f)|^2$ are flat. Verification via matlab is easy.

Oversampling

The principle of oversampling is simple: run ADC and DAC (Analog↔digital converters) at a sampling rate f_e well beyond Nyquist, say at $f_e = \beta f_0$ where $f_0 > 2B$ is a sufficient sampling rate.

We call $\beta > 1$ the *oversampling rate* (OSR); it is usually an integer. For audio CD, $\beta = 4$ (data is sampled at $f_e =$176'100 samples/sec but stored on the disc at $f_0 =$ 44'100 samples/sec).

To understand the benefits of oversampling, one has to recall the typical schema [link] of digital signal processing (DSP). First, in the ADC and DAC parts oversampling reduces the losses of quality due to working too close at Nyquist rate: sampling and reconstruction occur with a cutoff-frequency f_e far from the Nyquist rate $2B$ allowing for simple filter design or even omission of filtering. Second, as an additional benefit, oversampling reduces noise created by quantification, sampling or other sources of errors.

Note: down- and upsampling before and after processing (DSP: e.g.: de-noising, detection, enhancement, storage, transmission) allows for efficient computation and/or transmission at low rate.

Equation:

$$band - limit \rightarrow ADC \rightarrow \downarrow M \rightarrow DSP \rightarrow \uparrow M \rightarrow DAC \rightarrow smooth$$

Noise reduction under oversampling

In the context of oversampling, the signal has been sampled (and thereby quantized) at $f_e = \beta f_0$, with $f_0 > 2B$ and OSR $\beta >> 1$. Thus, we may decimate the quantized signal by a factor β and still keep the full signal quality. Decimation consists of digital low-pass filtering at cutoff frequency $1/(2\beta)$ (which corresponds to $f_0/2$), followed by downsampling.

Low-pass filtering and downsampling will not change the signal since $f_0 > 2B$, and thus decimation will not change the signal power (cpre. [link], [link], [link] magenta). However, *low-pass filtering will reduce the power of the noise*; the factor of reduction is β if an ideal filter is used. We offer several arguments for this fact.

Spectral picture of noise reduction (ideal filter). During the first step, low-pass filtering, the high frequencies of the power spectrum of the noise are cut away:

Equation:

$$S\left(f\right)=P\cdot \mathrm{Rect}\left(f\right)\rightarrow \cap \frac{1}{2\beta }\ \mathit{ideal\ filter\ sinc}\rightarrow {S}_{\mathrm{ideal}}\left(f\right)=P\cdot \mathrm{Rect}\left(f\beta \right)$$

After low-pass filtering, the noise is correctly band-limited (at a loss of power); consequently, downsampling will not change the power any further.

Spectral picture of noise reduction (digital filter). When using a digital filter $b=\left({b}_{1},\dots ,{b}_{q}\right)$ with energy ${E}_{b}={b}_{1}^{2}+\dots +{b}_{q}^{2}$ to filter the samples ${\varepsilon }_{n}$, the spectrum will change through filtering roughly as

Equation:

$$S\left(f\right)=P\cdot \mathrm{Rect}\left(f\right)\rightarrow \cap \frac{1}{2\beta }\mathit{digital\ filter}\,b\rightarrow {S}_{\mathrm{digital}}\left(f\right)\simeq P\cdot {E}_{b}\cdot \beta \cdot \mathrm{Rect}\left(f\beta \right)$$

Here, we used that the DFT of a well-designed digital filter b is roughly a rectangle of width $1/\beta$, meaning that a fraction $1/\beta$ of the samples ${\hat{b}}_{k}$ at the center take some constant value λ, the others are zero. It is easy to verify that $\lambda = \quad \overline{{E}_{b}\beta }$ in order for the energy of b to be ${E}_{b}$ (see Comment 8). Consequently, the portion $1/\beta$ of the K samples ${\hat{\varepsilon }}_{n}$ will be multiplied by λ, the others will be set to zero. Using [link] we see that $S(k/K)$ is multiplied by ${\lambda }^{2}={E}_{b}\beta$ for $-K/(2\beta)\le k\le K/(2\beta)$, and set to zero otherwise.

Comment 8 Let us denote by q the number of samples of the DFT of b. For a well designed digital filter there are roughly q/β samples equal to some constant λ, the remaining samples are zero. Since the DFT increases energy by factor equal to the sample size q, we get

Equation:

$$q{E}_{b}={\hat{b}}_{1}^{2}+\dots +{\hat{b}}_{q}^{2}={\lambda }^{2}q/\beta$$

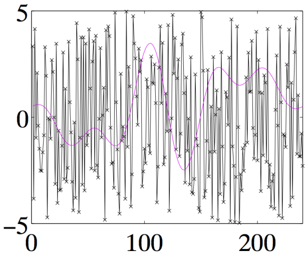
$\lambda = \quad \overline{{E}_{b}\beta }$.

According to [link] we have ${E}_{b}=q{P}_{b}\simeq 2{f}_{c}=1/\beta$ for a digital filter that is well designed, meaning that $c\simeq 1$. Thus, such a filter achieves approximatively the same power reduction as the ideal filter. Typically, however, ${E}_{b}$ is somewhat smaller than $1/\beta$.

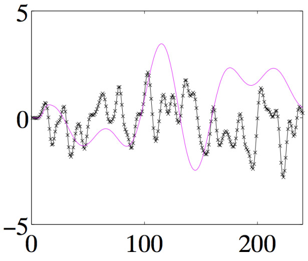
Noise samples

Black: White Noise under Decimation.
Magenta: For comparison, a signal with bandlimit B=0.0208 is subjected to the same manipulations as the Noise.
Details: $\Delta = 10$, $P = \Delta^2/12 = 8.3$, $M = 5$, $f_e = 1$, $f_d = 1/5 = 0.2$, $E_b = 0.16$ (length $q = 20$).

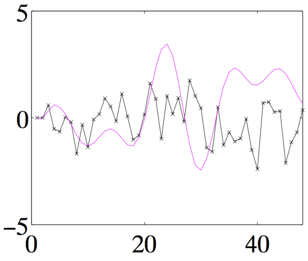
Note that the noise overwhelms the signal in rows 1 and 4; in rows 2 and 3 signal and noise are comparable. (The amplitude of the signal is kept unnaturally small for reasons of visualization).



White Noise ε_k

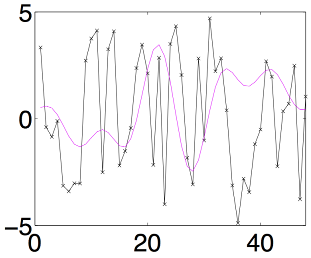


ε_k after lowpass filtering $\cap \frac{1}{2M}$



ε_k filtered and

downsampled

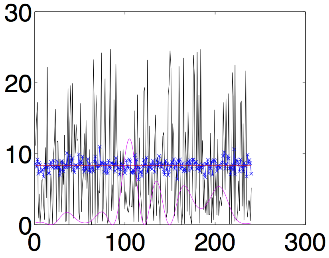


ε_k only downsampled.

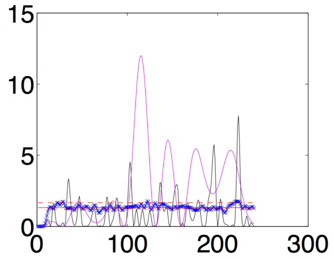
Noise samples squared

Black White Noise under Decimation. (details see figure 1)
Magenta: Signal for comparison,
Blue: the mean over 100 realizations of the noise.
Red: theoretical power; dashed with an ideal filter [link], solid with a digital filter [link].

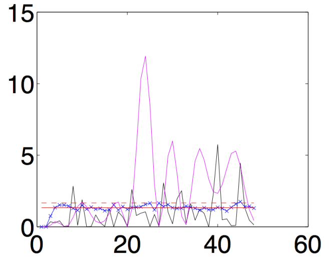
Note: the average of the squared noise terms (blue) is roughly constant at a level roughly equal to its power (red).



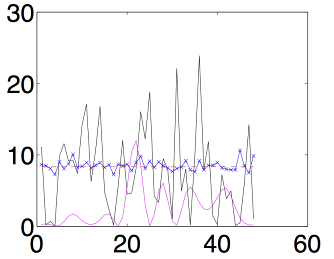
White Noise
$\varepsilon_k := 8.3$



after lowpass:
$P_{\text{ideal}} = P/M = 1.66$



lowpass and downsampled:
$P_{\text{ideal}} = P/M = 1.66$

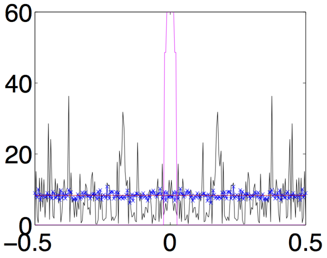


ε_k only downsampled;
$\varepsilon_k := 8.3$

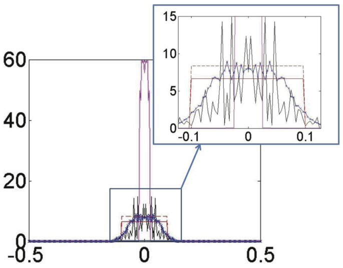
Estimation of S(f) by squared DFT

Black: $|\text{DFT}|^2/\text{K}$ of White Noise under Decimation (details see figure 1).
Magenta: Signal for comparison.
Blue: the mean over 100 realizations of the noise.

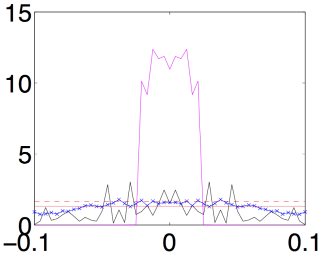
Red: theoretical power and S(f); dashed with an ideal filter [link], solid with a digital filter [link].
Note: the signal and its power are not changed because of band-limitation. The signal to noise ratio improves by a factor M when lowpass-filtering and downsampling (see (c)); it does not improve when only downsampling (see (d))



White Noise:
$S(f) = P = 8.3$
flat

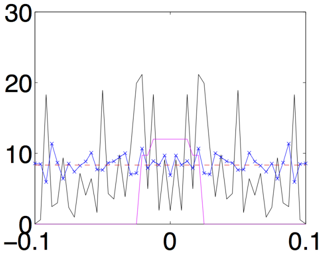


after lowpass: $S\,(f)$
cut off at
$1/2\mathrm{M} = 0.\,1,$
$\mathrm{P}_{\mathrm{ideal}} = \mathrm{P}/\mathrm{M} = 1.66$



lowpass and downsampled:
$S\,(f)\mathrm{P}_{\mathrm{ideal}} = \mathrm{P}/\mathrm{M} = 1.66$

flat



only downsampled:
$S\left(f\right)\mathrm{P}=8.3$ flat

Temporal picture of noise reduction. One may argue that only the high frequencies of the noise are removed, which are not used anyway. In an audio signal, for instance, setting $f_e=\beta\cdot 44kHz$ and decimating by β removes only the noise portion in the frequency range which can not be heard anyway. So, no benefit should result. This argumentation does not take into account that the power of noise is reduced which must result in smaller noise values. Comparing rows 2 and 4 in [link], we note that the noise magnitude has indeed decreased after filtering, but not under pure downsampling. Comparing the spectra $S(f)$ in [link] we note in the low frequencies from -0.1 to 0.1 the same average level of the spectral samples of roughly $P=\Delta^2/12=8.3$; however, for the downsampled noise all spectral samples are at this average level, while the filtered noise possesses $\beta=M$ more samples and only a portion $1/M$ of them are that level, the others are at zero. Thus, the power of the filter noise is M times smaller than the power of the downsampled noise. (This despite the fact that only high-frequency noise has been removed.)

Finally, we may also argue directly in the time domain using statistics. After filtering, the noise terms are $n_k=b_1\varepsilon_{k+1}+b_2\varepsilon_{k+2}+\ldots b_q\varepsilon_{k+q}$. Intuitively, this is some sort of average of the original noise, since $b_1+..+b_q=1$. But an average is always closer to the theoretical mean than the samples themselves, and the mean is here zero. So, intuitively, the filtered noise n_k takes smaller values. Since the mean of the filtered noise terms n_k is zero, a meaningful measure of their size is their power, i.e., their variance (cpre. [link]).

Using simple rules from statistics[footnote] we get $\mathbb{E}\left[n_k\right]=\ \ b_1^2+\ldots+b_q^2\ \ \mathbb{E}\left[\varepsilon\right]=E_b\cdot P$. Short: the noise level (its power) reduces by a factor E_b, just as we have seen before.

The variance of a sum of independent random variables is the sum of the variances; when multiplying a random variable by a constant b the variance in multiplied by b^2.

Illustration of noise reduction.

For a illustration see [link], [link], [link]. The main comparison is row 3 (oversampled and decimated) with row 4 (sampled at lower frequency). While the signal is practically identical, the noise is reduced dramatically in row 3.

Note also the dependence in the noise after lowpass filtering: it is possible to predict a few of the following noise terms. Indeed, the noise is not white and its spectrum is not flat. This dependence disappears almost entirely after downsampling: the prediction does not reach M samples into the future.

Note the shape of the estimated S(f) is not exactly a rectangle because the filter is not ideal (length 25). A longer filter would result in better shapes but also in longer delay.

Conclusion In summary, low-pass filtering reduces power by a factor β: Only a portion $1/\beta$ of the spectrum remains roughly unchanged the rest is set to zero. A formal calculation goes as follows, setting $f_e = 1$ as usual for digital filtering and $P = \Delta^2/12$ for a quantization noise (why we are interested in noise, and not in the noisy signal, see Comment 9)

Equation:

$$P_{\text{over}} = {}_{-1/2}^{1/2} S_{\text{filtered}}(f)df = {}_{-1/(2\beta)}^{1/(2\beta)} Pdf = \frac{1}{\beta}P = \frac{1}{\beta}\frac{\Delta^2}{12}$$

Comment 9 We do not compare power of signal to power of noisy signal, because they are almost equal; also, a theory of the power of the noisy signal is hard to develop since power(signal+noise) is not power(signal)+power(noise), unless signal and noise are uncorrelated, e.g. if the signal changes slowly and the noise has mean 0. In conclusion, we compare the power of the signal to the power of the noise. The ratio is called SNR “Signal to Noise Ratio”.

Since downsampling of a properly band-limited signal does not change its power (see [link], [link], [link]), the factor $1/\beta$ is the overall change of power during decimation. In other words, the SNR[footnote] improves under oversampling with subsequent decimation by β (which includes low-pass filtering at $1/2\beta$!) by the factor β, i.e., it improves by the oversampling rate (OSR). For a statistical argumentation, see box Comment 10.

The Signal to Noise Ratio (SNR) is the ratio of the power of signal and power of noise. Using Parseval's equation, it can be computed in the time or frequency domain:

Equation:

$$\text{SNR} = \frac{x(t)\ ^2 dt}{n(t)\ ^2 dt} = \frac{X(f)\ ^2 df}{N(f)\ ^2 df}$$

.

Comment 10 Statistical explanation of noise reduction. Short: filtering is a special way of averaging the samples. Averaging reduces variance, thus it reduces the noise power. In addition, the averaging is done in a special way as to leave small frequencies, i.e. slowly varying components, intact. Thus, the averaging reduces only noise, which is high frequency, i.e. quickly varying, and leave the signal intact. Somewhat more rigorously: Filtering noise means to compute

Equation:

$$\varepsilon'_k = \overset{M}{\underset{n=1}{}} b_n \varepsilon_{k-n}$$

Computing the variance of the resulting, filtered noise, we find:

Equation:

$$\text{var}\ \ \varepsilon'_k\ \ = \overset{M}{\underset{n=1}{}} b_n^2\, \text{var}\left(\varepsilon_{k-n}\right) = \sigma^2 \overset{M}{\underset{n=1}{}} b_n^2.$$

Here, we use that the original noise is independent. See footnote [link]. Now, for a FIR filter b it is easy to verify, e.g. with matlab, that $\overset{M}{\underset{n=1}{}} b_n^2 \simeq 1/\beta$. This approximation becomes more accurate the larger M is; it reflects the fact that the power of b is close to the power of the ideal lowpass filter c and the power of c is exactly $1/\beta$. A similar approximation holds also for any least square filter a: This is due to the fact that a approximates the ideal lowpass filter c in the least square sense. More precisely, let us denote the norm of c, i.e. the “length” of c by $\parallel c \parallel$. A well known formula says that $\parallel c \parallel = \ \ \overline{c_1^2 + ... + c_M^2}$. To approximate c in the least square sense means that $\parallel a - c \parallel$ is made as small as possible. But another well known fact says that $| \parallel a \parallel - \parallel c \parallel\ \ | \leq \parallel a - c \parallel$. This means, that the norms of a and c, thus their energies, are almost identical. In summary,

Equation:

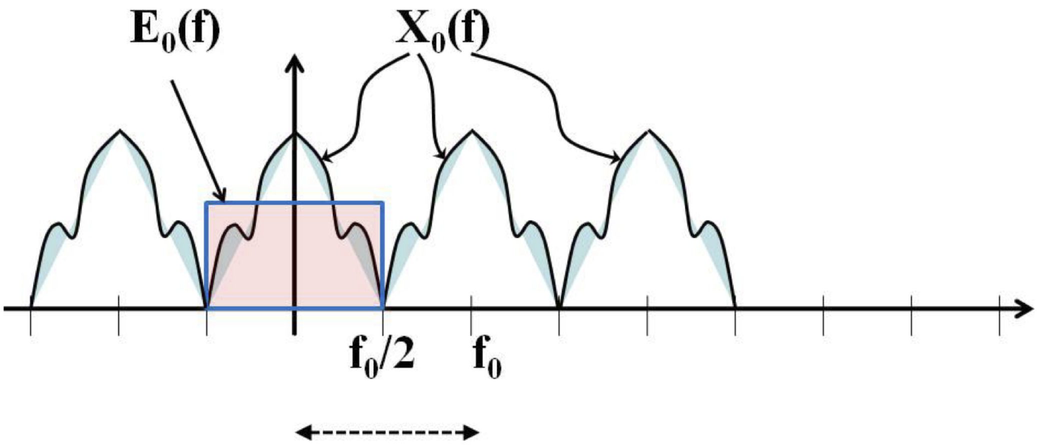
$$\text{var}\ \ \varepsilon'_k\ \ = \sigma^2/\beta.$$

In other words, the average power per sample is reduced be a factor β.

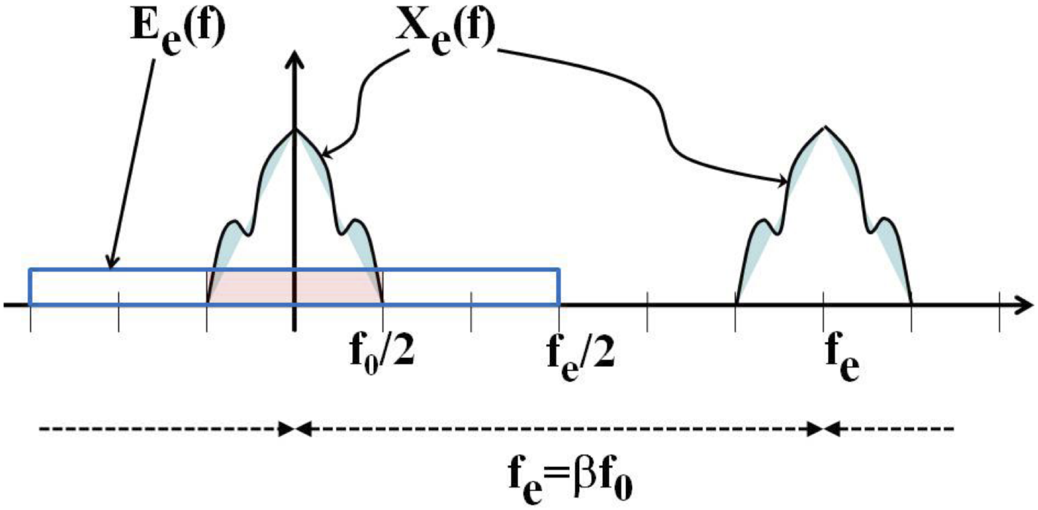
Noise-Shaping

Noise-shaping

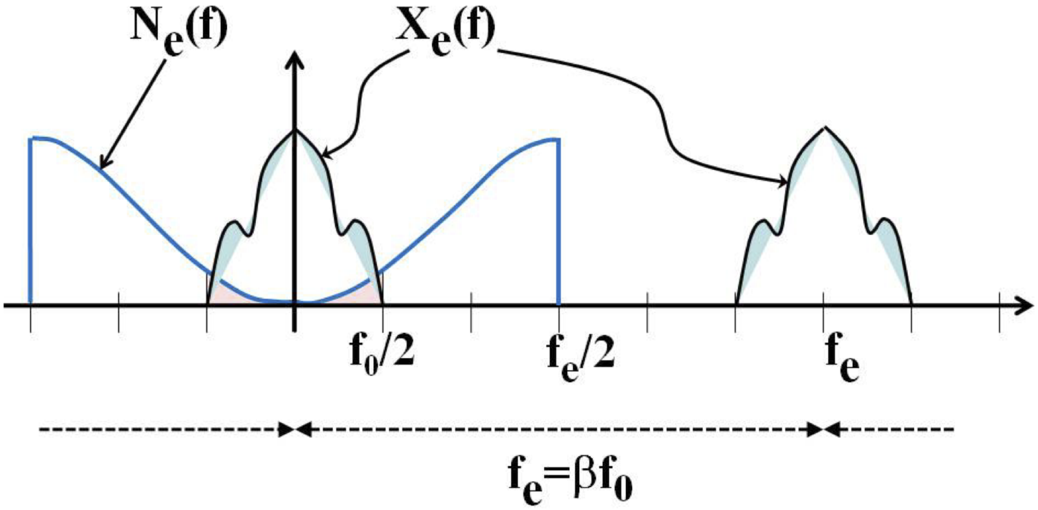
Noise shaping is a procedure which allows to put some color on an additive white noise. While this adds to the overall noise, one can shape the resulting noise such that its prevalent “color” lies in the high frequencies and at the same time reduce the presence of noise in the low frequencies.



Usual quantization noise.



When Oversampling, only the part in the band $[-f_0/2, f_0/2]$ will be relevant after down-sampling.



Noise Shaping moves the noise away from the smaller frequencies.

In the context of oversampling, noise shaping achieves an improvement if it is done so that it reduces noise in the band $[-f_0/2, f_0/2]$ and pushes it to the “colors” with frequencies in $[f_0/2, f_e/2]$ and $[-f_e/2, -f_0/2]$.

The most simple version of noise shaping results in two noise terms, the one produced by the system, one added with a delay on purpose. Set $\varepsilon_0 = 0$, then:

Equation:

$$\begin{aligned} x_k &\rightarrow -\varepsilon_{k-1} \rightarrow (x_k - \varepsilon_{k-1}) \rightarrow \\ &\rightarrow Quant. \rightarrow \begin{cases} y_k & \text{output} \\ \varepsilon_k = y_k - (x_k - \varepsilon_{k-1}) & \text{feedback} \end{cases} \end{aligned}$$

This assumes that we know or can know the error. Such is the case with quantization, where we can compute ε_k simply as the difference between output and input of the quantization (see [link]). This is not feasible in the example of a wireless channel with inference.

To study the effect of noise shaping we compute the Fourier transform of the overall error $n_k = y_k - x_k$. We find

Equation:

$$n_k = y_k - x_k = \varepsilon_k - \varepsilon_{k-1}$$

or

Equation:

$$n(t) = y(t) - x(t) = e(t) - e(t - \tau_e)$$

with Fourier transform (use that $f_e = 1/\tau_e$)

Equation:

$$\begin{aligned} N(f) &= E(f) - E(f) e^{-j2\pi\tau_e f} = E(f) \quad 1 - e^{-j2\pi f/f_e} \\ &= E(f) 2 \sin(\pi f/f_e) e^{-j\pi f/f_e} \end{aligned}$$

and power spectrum, (use that $2\sin^2(x) = 1 - \cos(2x)$)

Equation:

$$|N(f)|^2 = |E(f)|^2 2 \left(1 - \cos\left(\frac{2\pi f}{f_e}\right)\right)$$

Note that the spectrum is no longer flat; in other words, the noise n_k is colored. We note that the colored spectrum is small for small f, and it is still spread over a period of length f_e. Thus, noise shaping results in a reduction of noise in the small frequencies (see [link]).

Now we continue as with simple oversampling: since the signal has been sampled at $f_e = \beta f_0$, with $f_0 \geq 2B$ and $\beta > 1$ we may low-pass filter after noise shaping at cutoff frequency $f_0/2$. Thus, we take advantage of the fact that most of the power of the “shaped” or “colored” noise is in the `high'-frequency bands $[f_0/2, f_e/2]$ and $[-f_e/2, -f_0/2]$ with very little noise at frequencies around 0.

To assess the gain we compute the power of the noise after this low-pass filter. Note that the low-pass filter sets $N(f) = 0$ for $f_e/2 > |f| > f_0/2$: this removes much of the noise as demonstrated in the following computation; it also guarantees that power won't be changed when downsampling after filtering to f_0. Using again that $E(f) = \varDelta^2/12$ we get:

Equation:

$$\begin{aligned} P_{\text{shaping}} &= \frac{1}{f_e} {}_{-f_0/2}^{f_0/2} |N(f)|^2 df = \frac{1}{f_e} 2 {}_{0}^{f_0/2} |E(f)|^2 2\left(1- \cos\left(\frac{2\pi f}{f_e}\right)\right) \\ &= \frac{4\varDelta^2}{12 f_e}\left(f - \frac{f_e}{2\pi} \sin\left(\frac{2\pi f}{f_e}\right)\right)_{0}^{f_0/2} = \frac{\varDelta^2}{12} 4\left(\frac{1}{2\beta} - \frac{1}{2\pi} \sin\left(\frac{\pi}{\beta}\right)\right) \end{aligned}$$

Using the approximation $\sin(x) \simeq x - x^3/6 + x^5/5! - +\ldots$ we find

Equation:

$$\frac{P_{\text{shaping}}}{P_{\text{noise}}} = 4\left(\frac{1}{2\beta} - \frac{1}{2\pi} \sin\left(\frac{\pi}{\beta}\right)\right) \simeq 4\left(\frac{1}{2\beta} - \frac{1}{2\pi}\left(\frac{\pi}{\beta} - \frac{\pi^3}{6 \cdot \beta^3}\right)\right) = \frac{\pi^2}{3}\beta^{-3}$$

Noise reduction under oversampling with noise-shaping

In conclusion, the SNR improves (as compared to no oversampling) under noise shaping by the inverse of the factor [link], thus roughly by $\frac{3}{\pi^2}\beta^3$. In decibel, this corresponds to approximatively $10\log_{10}(3) - 20\log_{10}(\pi) + 30\log_{10}(\beta)$, thus, roughly 3 times more than with oversampling alone.

Numerical Examples:

With an OSR of $\beta = 128$ the SNR improves

- under oversampling alone by $10\log_{10}(128) \simeq 21\text{dB}$
- under oversampling coupled with basic noise-shaping as above by approximatively $10\log_{10}(3) - 20\log_{10}(\pi) + 30\log_{10}(128) = 58.0445\text{dB}$, or roughly 3 times more.

When doubling the OSR the SNR improves

- under oversampling alone by $10\log_{10}(2) \simeq 3.01\text{dB}$
- under oversampling with noise-shaping by approximatively $30\log_{10}(2) = 9.03$ dB, or approximatively 3 times more.